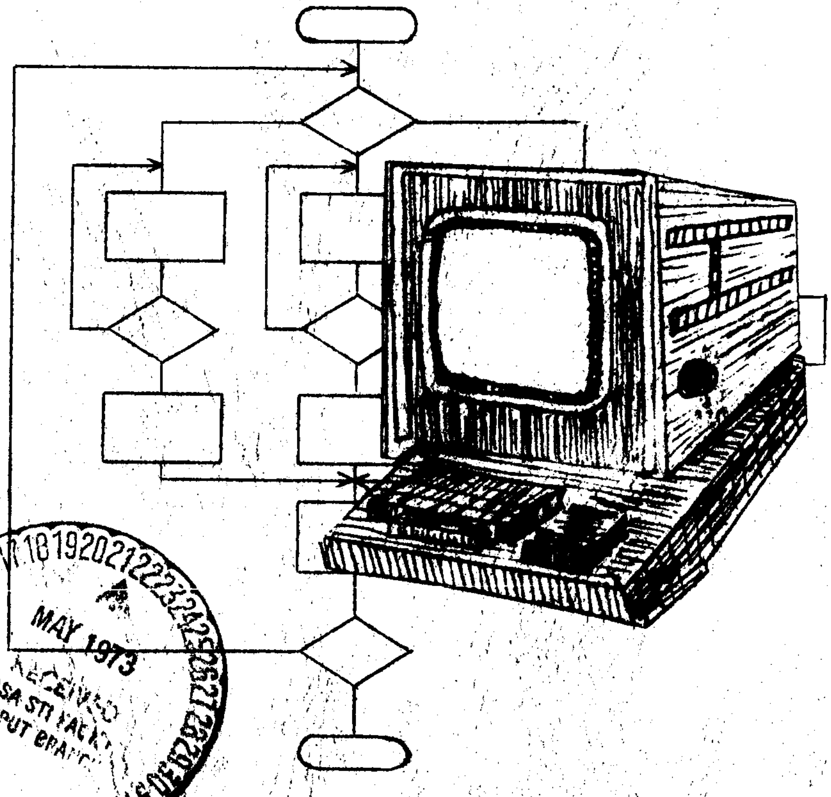
NASA CR 130231

NASA
Automatic System for Computer Program Documentation

VOLUME 1

FINAL REPORT



TEXAS A&M UNIVERSITY

DECEMBER 1972

AUTOMATIC SYSTEM FOR COMPUTER PROGRAM DOCUMENTATION

Dick B. Simmons, Roger W. Elliott,
Susan Arseven, and Daniel Colunga

Final Report

Contract NAS5-11911

December 1972

Submitted to

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771

Submitted by

Texas Engineering Experiment Station
Texas A&M University
College Station, Texas 77843

FOREWORD

This report presents the results of the project to design an automatic system for computer program documentation. This work was performed by the Texas Engineering Experiment Station at Texas A&M University, College Station, Texas. This work was performed under Contract NAS5-11911 for the National Aeronautics and Space Administration, Goddard Space Flight Center, Greenbelt, Maryland. The project monitor was Mr. E. P. Damon.

The authors wish to express their appreciation for the assistance provided by Randy Birge, Hank Goggan, Melvin McKinney, George Nichols, Ralph F. Planthold, Charles Schroeder, Andrew Sobey, Jr., Ernest R. Story, James R. Thames, and Darrell Ward.

This report was submitted on December 15, 1972.

ABSTRACT

This report describes the work done on a project to design an automatic system for computer program documentation. An extensive survey of documentation aids was made to determine what existing programs could be used effectively to document computer programs. Results of the study are included in the form of an extensive bibliography and working papers on appropriate operating systems, text editors, program editors, data structures, standards, decision tables, flowchart systems, and proprietary documentation aids. The preliminary design for an automated documentation system is included. An actual program has been documented in detail to demonstrate the types of output that can be produced by the proposed system.

TABLE OF CONTENTS

1.0 INTRODUCTION

Program documentation consists of all those items produced to help someone understand a program but which are not actually needed to produce the program. A program listing, when properly prepared, contains much descriptive information in addition to what is necessary to create the program. Documents such as written program descriptions, two dimensional data layouts, flowcharts, and cross references are all part of program documentation.

There is ample evidence that indicates the programmers, when properly motivated, are more than willing to provide needed program documentation. The problem of documentation is less one of ability to document than it is to define realistic documentation requirements. And, it is a general misconception that most programmers prefer to do things in their own way instead of in a standard way. Programmers are quite willing to follow a prescription for documentation as long as the policy is sound. A realistic management policy makes the difference between good and poor documentation.

To be generally useful, an automatic system for computer program documentation should document programs written in compiler or assembler languages. It should be independent of any manufacturer's computer or peripheral equipment. It should be applicable to a wide variety of types and sizes of programs. And, it should be capable of providing complete documentation for large,complex, high-use programs and more elementary documentation for smaller low-usage programs.

Such a system will be described in the following section. In general, documentation will be derived from information programmers normally supply when

designing, programming, and debugging a program. The system is both economical to operate and easy to use. The automated system described in this report allows a manager to specify the type and degree of documentation he would like produced on a project. The system aids him in controlling programming projects during development. Documentation produced by the system has been designed with flexibility to meet the needs of various types of programming projects.

The development of the automated documentation system design was accomplished over a period of nine months. During the first part of the study existing documentation aids were surveyed and their effectiveness was evaluated. An extensive bibliography of documentation aids was developed which is included in Appendix A of this report. To aid in evaluating documentation aids, working reports in areas such as program editors, text editors, operating systems, proprietary systems, flowcharting systems, and decision tables were produced. Existing documentation standards were examined. Audio-documentation techniques were explored as a possible alternative to written documentation.

Upon completion of the study and evaluation phase, an automatic documentation system was designed. One major objective of the system was to include existing documentation aids whenever possible. The system has been designed to make maximum use of existing software and to minimize the amount of additional software that needs to be developed in order to have an efficient, easy-to-operate, user-oriented automated documentation system. The system produces documentation during the development of a project as well as final documentation needed to understand and maintain the programming system.

In the following section the documentation aids study and evaluation phase will be described in detail. Then an overview of the system design will be

given after which a sample output from the proposed system will be described. A section describing the deliverable items described in the original contract is included after the sample program.

2.0 DOCUMENTATION AIDS - STUDY AND EVALUATION

An extensive study of existing documentation aids was made. Computer abstracts, relevant computer literature, and government information services were used to locate pertinent documents. A key word in context (KWIC) program was used to produce the bibliography contained in Appendix A. The bibliography cites 149 documents. In the first section of the bibliography, documents are listed by author. Each entry contains the author's name, the title of the document and the document number. In the next section, the document titles are listed according to key word with the document number to reference the full entry in the next section. In the third section of the bibliography, each document entered in the bibliography is listed by document number. The complete entry contains the name of the author, title, source of the reference and a brief statement about the contents of the reference. The last section of the bibliography is the KWIC index of document titles.

In addition to surveying existing literature and government data bases, a number of documentation aids which are supplied commercially were examined. A working paper was written describing AUTOCHART, AUTODIAGRAMMER, AUTODOC, AUTOFLOW, COMCHART, DYNACHART, EASYFLOW, FACTS, FLOWGEN/F, FORFLO, QUICKDRAW, SUPEREF, and FORDOC. All of the existing documentation aids had obvious failings. Many were restricted to either a single language or a single machine. None produced all of the documentation required to document a computer program. Most were able to document only at compilation level in contrast to a load module level

or a total system level. Many were limited to a single output device. Existing documentation aids are as difficult to use as a major compiler. If the programmer were to make use of more than one proprietary documentation aid he would have to make a major effort to learn the detailed rules of each package. Even though proprietary systems are quite expensive, they do not meet the total documentation needs of a major computer user in that most do not even address themselves to the problems of dealing with the textual descriptions, data layouts, etc., that must be included in every set of documentation.

While NASA has computers from almost every computer manufacturer, a major part of program development is done on IBM, CDC, and Univac systems. A working paper contained in Appendix C describes options available to the user through compilers, assemblers, and linkage editor/loaders. The operating systems evaluated were IBM's OS/360, Univac's EXEC 8, and CDC's SCOPE3. The languages chosen for study to determine options available were PL/1, FORTRAN, COBOL, and the Assembly language appropriate for the specific machine.

In the design of the automated documentation system, a decision was made not to modify any compiler or operating system. All information required for documentation would be either obtained from output normally produced by the appropriate software or from a special program produced to recreate the information. While the operating systems developed for the different machines are not the same, many options are very similar. An automatic documentation system that is designed to be used on any of the machines should take advantage of the common options of the operating systems. Tables within the working paper on operating systems describe the options of the systems and show which options are similar for the various operating systems.

Audio techniques were examined to determine their effectiveness as a tool for documentation. An experiment described in Appendix D was conducted in which programs were documented using both written and audio techniques. No significant advantage to either mode of documentation was noted. Probably the most useful place for audio recording techniques would be in the capturing of information which is later transcribed to written text. Many programmers who do not like to write descriptions of programming systems are perfectly willing to dictate them into a recorder. The recorder description can then be transcribed and edited into high quality written descriptions of programs.

Since much of the final documentation of programming systems is written text, text editors must be considered a major documentation aid. Many text editors such as the interactive ATS text editor on the IBM 360/370 or the batch TEXT360 available on the same systems required many man years to develop. Systems such as text editors should be used as subsystems of any automated documentation system developed so that it is not necessary to reproduce the features already built into existing text editors. Appendix E is a working paper describing text editors. Included are APL Text Editor, MTST, ASTROCOMP, DATATEXT, EDIT, ED PROCESSOR, ATS, TEXT/360, HES, FRESS, REDIT/RUNOFF, NLS, (TNLS, DNLS) and TEXT. Included in the Appendix are descriptions of the various systems and a table comparing them. Some of these text editors would be excellent subsystems of an automated documentation system.

Program editors were also examined during the study and evaluation phase. In Appendix F the CMS, WYLBUR, QED, TECO, TVEDIT, EDIT, and ED on-line program editors were compared. Also the LIBRARIAN, SIMPLE, CLOT, PROGRAM/MANAGE, CFMS, PANVALET, PLUS D-A, SPLIS-II, IEBUPDAT, IEBUPDTE, and ED off-line program

editors were described and compared. Program editors are not as complex as text editors, and most of the features of the program editors are included as subsets of the features of text editors. The program editor function would be developed as a part of the software necessary to control an automated documentation system that encompasses a wide spectrum of documentation aids.

A report was written by George C. Nichols describing a number of existing flowcharting systems.* A number of articles have been published on the various flowcharting systems. The most comprehensive comparison of flowcharting systems was written by Ned Chapin. ** N. Chapin mentions in his book that of the flowcharting systems he has evaluated, none could draw the standard flowchart symbols recommended by the American Standard Association. A system developed by D. B. Simmons at Bell Telephone Laboratories is operational and can draw any flowchart symbol on any type of output device to document any language. A number of flowcharting systems exist which could be driven by an automatic documentation system.

Decision tables are sometimes used as a type of documentation. Sometimes the actual source program is automatically produced from a type of decision table. An evaluation of existing decision table systems was done by J. R. Thames.*** In most cases, decision tables are used to produce a program instead of vice versa. It is felt that for most programming languages, it is not appropriate to automatically produce decision tables from source codes.

---

* Automatic documentation: A State of The Art Report by George C. Nichols written in partial fulfillment of a Master of Science degree in Computer Science

** Chapin, N., Flowcharts, Auerbach Publishers, Princeton, 1971

*** J. R. Thames, Documentation: Justification and Decision Table Exemplification, Texas A&M University, Spring, 1972 prepared in partial fulfillment of requirements for the degree of Master of Computing Science.

In most programming languages, the data description is either implicitly or explicitly described in a linear manner. When someone works with a complex data structure, special documentation techniques such as a two dimensional description of a data structure is useful. At the present time, very little has been done to automatically document data structures of programs. A short report was written and is included in Appendix G describing characteristics of the data structures of the PL/1, FORTRAN, and COBOL languages. Additional work needs to be done in the area of automatically describing data structures.

A very important facet of program documentation involves the standards that are used. There is no set of standards for all documentation that is produced describing programs. Standards exist for flowcharts but in other areas there are no generally accepted standards, and local practices usually take their place. A number of local documentation standards from both government and industry were examined and a short report on documentation standards is included in Appendix H.

As a result of the study evaluation phase, it was determined that many of the existing software systems such as text editors, flowcharters, language processors, operating systems, and data description software can be used as modules of a total documentation system. A flexible user-oriented documentation system is described in the following section.

3.0 AUTOMATIC SYSTEM FOR COMPUTER PROGRAM DOCUMENTATION

An automatic documentation system has been designed to produce timely up-to-date documentation at a relatively low cost. The system will document any computer language and run on any hardware taking advantage of existing

documentation aids. The system will be easy to use and will place no restrictions on the programmer. A detailed proposal for implementing the automatic documentation system is included in Appendix I. A brief summary of the system features is given in the next section.

3.1 System Features

The automatic documentation system will have the following features:

1. Minimal programmer restrictions - The system will be able to produce detailed, detail-suppressed and global flowcharts, data layouts, overlay descriptions, special cross reference glossaries, etc., from any computer program. Programmers who write comments in a standard way and who write their program descriptions using an interactive interrogator will be able to produce all documentation automatically. The system will be designed so that if a program is developed outside the system it will be fairly easy to retrofit the program into the system for documentation maintenance.

2. Eliminate all redundant efforts - Documentation produced in an early phase of the design or implementation process can be reused later on in the development process.

3. No operating system modification - No modifications will be made to any operating system. Items of documentation interest will be obtained by scanning output produced by operating systems or by reproducing information found in the internal tables of the operating system.

4. Use of existing documentation aids - Existing documentation aids will be used as modules in the comprehensive automated system.
5. Interactive/batch system - An interactive mode with special features for interrogating the user of the system will be used to obtain necessary documentation aids. An alternative batch mode will be available.
6. Documentation during development - The documentation data base will be constructed from information gathered during all design phases. As the design progresses, the user supplies only the information not available from previous steps.
7. Accept any language - The automated documentation system will be language independent. Initially the system will be designed to accept FORTRAN, COBOL, PL/1, and Assembly Language.
8. Operate on any hardware - Initially the system will be designed to operate on the IBM 360 with planned expansion to the Univac 1108 and the CDC 6600 computer systems.
9. Monitor and control projects - Features will be designed into the system to allow the project manager to monitor the exact status of program development and program documentation. In addition, system access and security will be under his control.

3.2 Initial Phase

It is proposed that in January, 1973, initial development of the automatic system for computer program documentation will begin. A twelve month period beginning at that time will be termed the initial phase. During the initial phase either FORTRAN or COBOL will be used to write programs that make up the automatic

documentation system. Initial development will be for a system operational on an IBM 360 computer, but software will be easily transportable to the Univac or CDC system. The automatic documentation system will document COBOL, FORTRAN, PL/1, or assembler language programs.

A detailed description of how the automated documentation system can be used for documentation during development as well as for documentation when development is complete is described in the proposal contained in Appendix I.

3.3 System Structure

Three types of programs will be used in the system. Types 1 and 2 are new programs that must be developed. Type 3 programs are existing programs that can be used without change. Type 3 programs make up a major part of the software necessary to implement the automatic documentation system. New programs will not be developed when operational documentation aids are available. By doing this, a sophisticated system will be developed at a relatively low cost. A detailed description of the Type 1, Type 2, and Type 3 programs is contained in Appendix I. The detailed descriptions of the programs to be developed and the types of documentation produced by the system are described in Appendix I.

3.4 Advantages

Use of the proposed automatic documentation system offers many advantages over other techniques for developing programs and producing documentation. The system will be user-oriented and will be as easy to operate as existing on-line or batch program editors. Programmers who use good programming practices in developing software can use all of the system features without

extra effort. Programmer productivity will be enhanced by improved communications during the development process. A modular system will allow new types of documentation to be easily added to the system. This is the first system to bring together all types of documentation aids into a single system. It emphasizes documentation on a load module and system basis as well as for a single compilation. Documents can be produced containing heterogeneous outputs such as text and flowcharts. Managers can use the system to control and monitor projects. Program and documentation standards can be enforced and taught through the use of automatic documentation systems. People who do not use the documentation system during program development will be able to use it for post-development documentation. Programs developed independently of the automatic documentation system can easily be retrofited into the documentation data base.

4.0 SAMPLE PROGRAM DOCUMENTATION

A sample of the types of documentation to be produced by the automatic system is included in Appendix J. Since the types and formats of the documentation prepared is completely determined by the user when he specifies templates for data to be collected and the recipes for documentation to be produced, the sample is just one possible format that a manager could select. The templates and recipes can be varied from one project to the other.

The subject of the sample documentation was a set of fifteen Fortran programs called "DYNASOR-II, A Finite Element Program for the Dynamic Nonlinear Analysis of Shells of Revolution" developed for NASA by Joe R. Tillerson and Walter E. Haisler of the Aerospace Engineering Department, Texas A&M University. The DYNASOR II programs were developed to compute the nonlinear dynamic response of shells of revolution in relatively short periods of computer time for a large number of important shell problems.

The sample documentation for the DYNASOR II system is in three parts, a User's Manual, a Program Maintenance Manual and an Operations Manual. These manuals bring together information generated in various stages of the development process by such people as the specifier, designer, programmer and the validator. The sample documentation shows how the same information can be used to produce documentation for the system user, maintainer or operator.

The sample User's Manual is designed to explain to a scientist or engineer the problem solution and to show him how to prepare input to use the programs. It is prepared assuming that the user does not know computer programming.

The User's Manual begins with a title page, abstract, system overview, system flowcharts, and environment and configuration descriptions. These components can also be used without change in the maintenance and operator guides. The User's Manual then describes the method of analysis of the problem. Included next are guidelines for the user and program limitations. The next section covers the meaning of the input parameters and how to prepare the input. The solution of example problems using DYNASOR II are included to further clarify the system for the user. A list of references to the methods of analysis is given in case the user would like to study the techniques in detail. A special restart feature of the DYNASOR II program is explained in addition to a discussion on how to properly choose certain input parameters (in this case loads and temperatures) for best results. Functional flowcharts are included for the user interested in the structure of the programs, their functions and how they work together.

The Maintenance Manual is designed to familiarize the maintenance programmer with the set of DYNASOR II programs and to serve as his reference when modifying or correcting the programs. The maintenance manual is introduced by the title page, abstract, system overview, system flowchart and environment

description. The next sections contain detailed descriptions of the computer aspects of software being documented on system, global and local levels. On the system level there is a subroutine connectivity diagram, a system flowchart, and Job Control Language and deck setup descriptions. On the global level, overlay maps, global flowcharts, global data descriptions, control card descriptions and subroutine calls are included. On the local level, the Maintenance Manual contains detailed, detail suppressed, and functional flowcharts; local data descriptions; label cross references and individual program listings. Also included are system test samples and evaluation criteria. Other sections could be included here concerning such items as rules for programming practices, naming conventions, mathematical symbols and the like.

The Operator's Manual is designed to be used by the person who is responsible for running of the system on the computer. He is not required to have a knowledge of the scientific problem or of computer programming.

Again the manual begins with title page, abstract, system overview, system flowchart, and environment page duplicated automatically by the documentation system. The next sections describe components of the operation of the DYNASOR II runs. The purpose of each run is described along with its relationship to other runs, set up and run instructions, run frequency, run prerequisites, controls and schedules. Each data set is described along with the file characteristics. The job control language and control formats are detailed along with directions for setting up the deck.

Error messages are listed with explanations and actions to be taken. There is a section on checkpoint, restart, error procedures, backup and recovery procedures.

Any of the parts of any of the manuals in Appendix J can be produced separately at any stage of development where appropriate data is available in the data base.

5.0 DELIVERABLE ITEMS

Article II of NASA Contract NAS5-11911 requested the following deliverable items:

a) Items of interest for documentation purposes that can be obtained automatically by the use of specifically designed computer programs and/or modifications to the operating system and associated software.

   Output that can be obtained from commerical documentation aids is described in Appendix B. Text editors are described in Appendix C and program editors are described in Appendix F. No modifications to operating systems or associated software are recommended. Items of interest for documentation purposes that can be obtained automatically by the proposed automatic documentation system are described in Appendix I.

b) Sample formats of useful and alternate printouts applicable to program documentation; derived from (a).

   Sample outputs of the type of documentation that will be produced by the automatic documentation system are included in Appendix J.

c) Methods by which the necessary data are obtained from source decks and/or operating system components (compilers, loaders, etc.).

   A detailed description of the methods by which the necessary data are obtained from source decks and/or operating system components is described in the Appendix I.

d) Programming disciplines, restrictions or coding requirements imposed upon programmers in order to enable them to produce documentation by using software recommended as a result of this effort.

The report should indicate what behavior may be expected if required disciplines are not observed and obeyed. In establishing these constraints, consideration must be given as to what may be reasonably expected in the way of programmer cooperation.

The automatic documentation system will be as easy to use as the typical program editor. For programmers who use text editors in developing their program descriptions, additional restrictions will not be necessary to the textual part of the documentation. A large part of the documentation necessary to describe the program can be produced with no programmer restrictions as described in Appendix I. For such things as high quality functional flowcharts, the programmer will be required to stylize his program comments. A program developed outside the automatic documentation system can be retrofitted into the documentation system. In the design of the automatic documentation system, a major effort will be made to minimize restrictions on programmers.

e) Certain of the outputs desirable for program documentation purposes may be obtained by modifying standard software components. Inasmuch as most compilers, loaders, etc., extract relevant information and retain it internally, much of the program development needed to produce this data in readable form may be avoided; however, two points must be borne in mind. First, this effort is not directed toward a specific type or model of computer or software system; hence, the techniques employed must be more or less generally applicable. Second, it is most desirable that system modifications be kept to a minimum; hence, it would be preferable to confine such modifications

to a few system modules as is practicable and to modularize any proposed extensions to the fullest possible extent.

No standard software components will be modified.

f) If one or more specific operating systems are cited as illustrative examples, or for purposes of investigation, the particular modules or components of the system(s) requiring change, and the nature of such changes, should be fully described to the extent that they are penetrated by this study.

No modules or components of operating systems need be changed to work with the proposed automatic documentation system.

g) While certain information may be extracted by altering system components, certain independent programs may be required to complete the package in order to supplement, broaden and/or organize meaningful documentation elements. Such programs must be fully described, including inputs required, functions performed, and outputs derived.

Programs required to complete the package in order to supplement, broaden and/or organize meaningful documentation elements are Type 1 and Type 2 programs described in the proposal in Appendix I. Sample output provided by the system is described in Appendix J.

h) As a separate document or separate section of the final report, there will be a consolidation listing of all those techniques which could be implemented without changes in hardware or operating systems. As a subset of the above, there will be a designation of complexity level for the incorporation into a working program. Those which require little or no effort or change in style by the applications programmer should be specifically defined.

No changes will need to be made to program statements in a working program for documentation purposes. Documentation systems might make changes to the comment fields in a working program, but this will be done in a manner that will not, in any way, affect the other statements in operational programs. All noncomment program statements will be fully protected during any mode in which the program comments are being changed.

i) This section should be presented and recommended as an integrated, usable package which could be implemented immediately to provide a partial solution while further work is being done on the more elaborate improvements.

Commercially available documentation aids that can be used immediately are described in Appendix B. The initial phase of the automated documentation system to be implemented is described in Appendix I. Future elaborations and extensions to the system are also described in the proposal in Appendix I.

j) The outputs produced by the automated methods devised should contribute to final program documentation. Techniques should be recommended which best present a format for future computer program documentation.

A sample output of the type that would be produced by the automated system has been reproduced in Appendix J. This example demonstrates the type of output that would be produced by the system. The specific format of the manual desired by a project manager would be determined by the project manager.

APPENDIX A

BIBLIOGRAPHY

AIRES CORP.	
AUTODIAGRAMMER II	720614
AMERICAN NATIONAL STANDARDS INSTITUTE, INC.	
AMERICAN NATIONAL STANDARD FLOWCHART SYMBOLS AND THEIR USAGE IN INFORMATION PROCESSING	700905
ANDERSON, H. E.	
AUTOMATED PLOTTING OF FLOWCHARTS ON A SMALL COMPUTER	650101
APPLICATIONS PROGRAMMING CO.	
DYNACHART	720610
APPLIED DATA RESEARCH	
AUTOFLOW	720612
APPLIED DATA RESEARCH, INC.	
LIBRARIAN	720601
ARMERDING, G.W.	
THE JOSS PRIMER	670804
ARPA NETWORK INFORMATION CENTER	
NETWORK INFORMATION CENTER USER GUIDE	720701
AUERBACH ASSOCIATES, INC.	
DATA MANIPULATION	720603
AZZARI, A.I.	
DEVELOPMENT OF A COMPUTER PROGRAM FOR GENERATING TROUBLE-SHOOTING DECISION TREES	670901
BARNES, L.	
RUNOFF: A PROGRAM FOR THE PREPARATION OF DOCUMENTS	700301
BECKWITH, R.C.	
A TECHNIQUE FOR COMPUTER FLOWCHART GENERATION.	680801

BELL SYSTEMS PRACTICES	
PROGRAM DOCUMENTATION STANDARDS	660403
BELOIT COMPUTER CENTER	
AUTOCHART	720613
BERKELEY, E.C.	
RESEARCH IN COMPUTER-ASSISTED DOCUMENTATION OF COMPUTER PROGRAMS	701202
BLUMBERG, S.E.	
AN INTERIM PROGRESS REPORT OF COMPUTER OUTPUT MICROFILM ACTIVITIES AND EXPERIENCES	700701
BRACKETT, J.W.	
MAINTENANCE AND ENHANCEMENT OF THE AED SYSTEM	710602
BROOKS, A.I.	
DEVELOPMENT OF A COMPUTER PROGRAM FOR GENERATING TROUBLE-SHOOTING DECISION TREES	670901
BUSINESS AUTOMATION	
TALKING DOWN A PROGRAM	720501
CALCOMP	
FLOWGEN/F	720609
CALLAHAM, M.D.	
AUTODOC: COMPUTER-BASED ASSISTANCE FOR DOCUMENT PRODUCTION	670001
CANTRELL, H.N.	
LOGIC STRUCTURE TABLES	610601
CHAPIN, N.	
FLOWCHARTS	000104
PARSING OF DECISION TABLES	670803
PROGRAM DOCUMENTATION: THE VALUABLE BURDEN	680501
PERSPECTIVE ON FLOWCHART PACKAGES	710301

A-1-a

CHUBB, C.P.

DEVELOPMENT OF A COMPUTER PROGRAM FOR GENERATING TROUBLE-SHOOTING DECISION TREES 670901

COMMUNICATIONS COMPUTER PROGRAMMING CENTER

AUTOMATED WEATHER NETWORK 000101

COMP. SCI. DEPT. UNIVERSITY OF ILLINOIS

ILLIAC IV QUARTERLY PROGRESS REPORT 671201

COMPRESS, INC.

DOCUMENTATION SOFTWARE 670801

COMCHART 720605

CONKLIN, C.F.

EVOLUTIONARY SYSTEM FOR DATA PROCESSING, VOLUMES I-IV. 000102

CONTI, E.

STUDY OF MACHINE-AIDED POST EDITING 680202

CONTROL DATA CORPORATION

CONTROL DATA 6000 SERIES COMPUTER SYSTEMS REFERENCE MANUAL NUMBER 60100000 650701

CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COBOL REFERENCE MANUAL PUBLICATION NUMBER 60191200 690901

CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COMPASS REFERENCE MANUAL PUBLICATION NUMBER 60190900 691001

CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS FORTRAN REFERENCE MANUAL PUBLICATION NUMBER 60174900 691103

EDIT - KRONOS TEXT EDITOR (EDIT) REFERENCE MANUAL 710104

KRONOS TEXT EDITOR (EDIT) 710105

CULLINANE CORPORATION

PLUS D/A, A DIRECT ACCESS SOFTWARE SECURITY AND CONTROL SYSTEM 720201

DATA FOR MANAGEMENT DECISION

FORTRAN VARIABLE NAME DOCUMENTER 720607

DATA INSTRUMENTS CO.

AUTODOC 720615

DEFENSE DOCUMENTATION CENTER

BIBLIOGRAPHY ON INFORMATION SCIENCES. VOLUME I 680201

COMPUTER IN INFORMATION SCIENCES. VOLUME III 681002

MICROFICHE, MICROFILM, AND RELATED EQUIPMENT 711001

DEMUTH, N.

STUDY OF MACHINE-AIDED POST EDITING 680202

DIAL, R.B.

DECISION TABLE TRANSLATION 700902

DNA SYSTEMS, INC.

FORFLO 720608

ELLIS, T.O.

ARPA NETWORK SERIES I.INTRODUCTION TO THE ARPA NETWORK AT RAND AND TO THE RAND VIDEO GRAPHICS SYSTEM 710901

ENSLOW,JR. P.H.

DIAGNOSTIC PROGRAMMING FOR DIGITAL COMPUTERS A BIBLIOGRAPHY 670402

FELDMANN, C.G.

MAINTENANCE AND ENHANCEMENT OF THE AED SYSTEM 700904

FELSMAN, W.O.

COST ADVANTAGES OF AN INTEGRATED DOCUMENTATION APPROACH 701113

FERGUS, R.M.

AN INTRODUCTION TO DECISION TABLES 680701

IBM CORPORATION
CONTINUATION

360S-ED-510 360S-ED-521 360S-LD-547 — 691102

IBM SYSTEM/360 OPERATING SYSTEM FULL AMERICAN NATIONAL STANDARD COBOL PROGRAMMERS GUIDE NUMBERS 360S-CB-545 — 700602

IBM SYSTEM/360 OPERATING SYSTEM FORTRAN IV (G AND H) PROGRAMMERS GUIDE PROGRAM NUMBERS 360S-FO-500 360S-FO-520 — 700603

IBM SYSTEM/360 OPERATING SYSTEM ASSEMBLER F PROGRAMMERS GUIDE PROGRAM NUMBER 360L-AS-037 — 701203

IBM SYSTEM/360 OPERATING SYSTEM PL/I(F) PROGRAMMERS GUIDE PROGRAM NUMBER 360S-NL-511 — 710101

INFORMATION CONTROL SYSTEMS,INC.

ASTROCOMP — 720702

INTERACTIVE SCIENCES CORPORATION

TECO - TEXT EDITOR AND CORRECTOR - REFERENCE MANUAL — 710603

JONES, M.V.

COMPUTERIZING A GOVERNMENT DATA SYSTEM: A MANAGEMENT OVERVIEW OF THE STEPS REQUIRED AND THE TIME NEEDED. — 671202

KALAR, M.C.

COSMIC: PROGRAM DOCUMENTATION EXPERIENCE — 701111

KAVANAGH, T.E.

USING DECISION STRUCTURE TABLES — 640301

KEIRSTEAD, R.E.

ON THE FEASIBILITY OF FORMAL CERTIFICATION — 000103

KING, F.E.

LOGIC STRUCTURE TABLES — 610601

KING, P.J.H.

CONVERSION OF DECISION TABLES TO COMPUTER PROGRAMS BY RULE MASK TECHNIQUES — 661101

KING, P.J.H.
CONTINUATION

DECISION TABLES — 670802

AMBIGUITY IN LIMITED ENTRY DECISION TABLES — 681001

THE INTERPRETATION OF LIMITED ENTRY DECISION TABLE FORMAT AND RELATIONSHIPS AMONG CONDITIONS — 691101

KIRK, H.W.

USE OF DECISION TABLES IN COMPUTER PROGRAMMING — 650102

KNUTH, D. E.

COMPUTER-DRAWN FLOWCHARTS. — 630901

KREUTZER, P.J.

DATA COMPRESSION IN LARGE BUSINESS-ORIENTED FILES — 711002

KRIDER, L.

A FLOW ANALYSIS ALGORITHM — 641001

LANZANO, B.C.

PROGRAM AUTOMATED DOCUMENTATION METHODS — 701109

LOBACK, L.R.

S/360 ASSEMBLER/COMPILER LISTINGS LIBRARY MAINTENANCE, NO. 360D-0C.0.015 — 681101

LONDON, K.R.

DOCUMENTATION STANDARDS — 690103

LUDWIG, H.R.

SIMULATION WITH DECISION TABLES — 680101

LUEBKE, W.R.

THE INTEGRATION OF SYSTEM SPECIFICATIONS AND PROGRAM CODING. A REPORT ON EXPERIENCE OF THE MEDLARS II PROJECT. — 701114

MALTBY, Q.J.

CONTROL SECTION CROSS-REFERENCE AND LINK EDIT EDITOR — 681003

MANAGEMENT SYSTEMS CORPORATION

PROGRAM/MANAGE, A PROGRAM TO MAINTAIN SOURCE PROGRAM 720602

MANTECH CORPORATION

SUPEREF 720606

MARKS, S.L.

THE JOSS PRIMER 670804

MARTIN, W. W.

FLOWCHARTING: SHORTHAND, ANALYSIS AND MODEL. 660201

MCCLURE, C.W.

AUTOMATED ENGINEERING DESIGN AN APPROACH TO AUTOMATED DOCUMENTATION 701108

MEADOW, C.T.

EVOLUTIONARY SYSTEM FOR DATA PROCESSING, VOLUMES I-IV. 000102

MEL'CHUK, I.A.

AUTOMATION IN LINGUISTICS 680103

MENKUS, B.

DEFINING ADEQUATE SYSTEMS DOCUMENTATION 701201

MERCER, V.

AUTOCHART NEW METHOD CUTS MANUFACTURING. 640501

MILLS, H.D.

SYNTAX-DIRECTED DOCUMENTATION FOR PL360 700401

MONTALBANO, M.

TABLES, FLOW CHARTS, AND PROGRAM LOGIC 620901

MOUTON, M.L.

DOCUMENTATION: MOTIVATION AND TRAINING OR AUTOMATION 701102

MUTHUKRISHNAN, C.R.

ON THE CONVERSION OF DECISION TABLES TO

MUTHUKRISHNAN, C.R.
CONTINUATION

COMPUTER PROGRAMS 700601

NATIONAL COMPUTER ANALYSIS, INC.

COBOL FLOWCHARTING 680301

QUICK-DRAW 720611

NATIONAL MILITARY COMMAND SYSTEM SUPPORT CENTER COMPUTER

USER'S MANUAL NMCS INFORMATION PROCESSING SYSTEM (NIPS) (IBM 1410/7010) 680102

NEELY, M.D.

AUTOMATIC PROGRAM ANNOTATION (AUTONOTE) 701104

NEPERLD, R.

MICROFICHE STORAGE AND RETRIEVAL SYSTEMS STUDY: 700801

NINKE, W. H.

THE GROWTH OF COMPUTER GRAPHICS AT BELL LABORATORIES. 680601

O'BRIEN, F.

A TECHNIQUE FOR COMPUTER FLOWCHART GENERATION. 680801

OPERATIONS DIVISION STRATEGIC SYSTEMS DEPARTMENT

AUTOMATIC DATA PROCESSING SUPPORT FOR FLAG PLOT 660402

PANSOPHIC SYSTEMS, INC.

PANVALET - THE DIRECT ACCESS LIBRARY SYSTEM 720502

PARDEE, S.

BELLFLOW AUTOMATIC FLOWCHARTING SYSTEM 701105

PARKER, D.B.

ON THE FEASIBILITY OF FORMAL CERTIFICATION 000103

PENSIERO, W.S.

SLIP - SOURCE LIBRARY INQUIRY PROGRAM, NO.

PENSIERO, W.S.
CONTINUATION

3600-00.3.009 660601

POLLACK, S.L.

CONVERSION OF LIMITED ENTRY DECISION
TABLES TO COMPUTER PROGRAMS 651101

REAM, N.J.

ON-LINE MANAGEMENT SYSTEMS INFORMATION 690401

REINWALD, L.T.

CONVERSION OF LIMITED-ENTRY DECISION
TABLES TO OPTIMAL COMPUTER PROGRAMS I:
MINIMUM AVERAGE PROCESSING TIME 660801

CONVERSION OF LIMITED-ENTRY DECISION
TABLES TO OPTIMAL COMPUTER PROGRAMS II:
MINIMUM STORAGE REQUIREMENT 671102

RICE, D.E.

ON-LINE TEXT EDITING: A SURVEY 720604

RICH, R.P.

AUTOMATIC EDITING OF MANUALS 701112

ROSEN, M.H.

S/360 ASSEMBLER/COMPILER LISTINGS LIBRARY
MAINTENANCE, NO. 3600-00.0.015 681101

SALISBURY, A.B.

DIAGNOSTIC PROGRAMMING FOR DIGITAL
COMPUTERS A BIBLIOGRAPHY 670402

SCHWAYDER, K.

CONVERSION OF LIMITED-ENTRY DECISION
TABLES TO COMPUTE PROGRAMS A PROPOSED
MODIFICATION TO POLLACKS ALGORITHM 710201

SCOTT, A. E.

AUTOMATIC PREPARATION OF FLOWCHART
LISTING. 580101

SHERMAN, PHILIP M.

FLOWTRACE, A PROGRAM FOR FLOWCHARTING

SHERMAN, PHILIP M.
CONTINUATION

PROGRAMS. 661201

SHMIDT, D.T.

USING DECISION STRUCTURE TABLES 640301

SHOBER, J.A.

DECISION TABLES FOR BERRER MANAGEMENT
SYSTEMS 660301

SIGMA SOFTWARE COMPANY

CFMS - CARD FILE MAINTENANCE SYSTEM 720101

SNYDER, R.

PROGRAMMING DOCUMENTATION 651001

SOLAND, R.M.

CONVERSION OF LIMITED-ENTRY DECISION
TABLES TO OPTIMAL COMPUTER PROGRAMS I:
MINIMUM AVERAGE PROCESSING TIME 660801

SPERRY RAND CORPORATION

UNIVAC 1108 COBOL UP-4048 670101

UNIVAC 1100 SERIES OPERATING SYSTEM 710102

ED PROCESSOR - UNIVAC 1100 SERIES
OPERATING SYSTEM PROGRAMMER REFERENCE 710103

STANFORD UNIVERSITY COMPUTATION CENTER

WYLBUR 700903

STEINBERG, L.

DEVELOPMENT OF A COMPUTER PROGRAM FOR
GENERATING TROUBLE-SHOOTING DECISION TREES 670901

SWANSON, R.W.

INFORMATION SCIENCES: SOME RESEARCH
DIRECTIVES 660401

SYSTONETICS CORP.

EZFLOW 720616

ALGORITHMS

AUTOMATIC PREPARATION OF FLOWCHART LISTING.	580101
TABLES, FLOW CHARTS, AND PROGRAM LOGIC	620901
A FLOW ANALYSIS ALGORITHM	641001
USE OF DECISION TABLES IN COMPUTER PROGRAMMING	650102
CONVERSION OF LIMITED ENTRY DECISION TABLES TO COMPUTER PROGRAMS	651101
DATA, DOCUMENTATION, AND DECISION TABLES	660101
CONVERSION OF LIMITED-ENTRY DECISION TABLES TO OPTIMAL COMPUTER PROGRAMS I: MINIMUM AVERAGE PROCESSING TIME	660801
CONVERSION OF DECISION TABLES TO COMPUTER PROGRAMS BY RULE MASK TECHNIQUES	661101
DECISION TABLES	670802
PARSING OF DECISION TABLES	670803
DEVELOPMENT OF A COMPUTER PROGRAM FOR GENERATING TROUBLE-SHOOTING DECISION TREES	670901
CONVERSION OF LIMITED-ENTRY DECISION TABLES TO OPTIMAL COMPUTER PROGRAMS II: MINIMUM STORAGE REQUIREMENT	671102
SIMULATION WITH DECISION TABLES	680101
THE USE OF DECISION TABLES WITHIN SYSTEMATICS	680802
AMBIGUITY IN LIMITED ENTRY DECISION TABLES	681001
THE INTERPRETATION OF LIMITED ENTRY DECISION TABLE FORMAT AND RELATIONSHIPS AMONG CONDITIONS	691101
ON THE CONVERSION OF DECISION TABLES TO COMPUTER PROGRAMS	700601
DECISION TABLE TRANSLATION	700902
RESEARCH IN COMPUTER-ASSISTED DOCUMENTATION OF COMPUTER PROGRAMS	701202
UNIVAC 1100 SERIES OPERATING SYSTEM	710102

ALGORITHMS CONTINUATION

CONVERSION OF LIMITED-ENTRY DECISION TABLES TO COMPUTE PROGRAMS A PROPOSED MODIFICATION TO POLLACKS ALGORITHM	710201
A LIBRARY OF DATA COLLECTION AND MANIPULATION SUBROUTINES	710902

ASSEMBLER-ROUTINES

ILLIAC IV QUARTERLY PROGRESS REPORT	671201
CONTROL SECTION CROSS-REFERENCE AND LINK EDIT EDITOR	681003
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COMPASS REFERENCE MANUAL PUBLICATION NUMBER 60190900	691001
IBM SYSTEM/360 OPERATING SYSTEM ASSEMBLER F PROGRAMMERS GUIDE PROGRAM NUMBER 360L-AS-037	701203

COMPILERS

LOGIC STRUCTURE TABLES	610601
UNIVAC 1108 COBOL UP-4048	670101
THE GROWTH OF COMPUTER GRAPHICS AT BELL LABORATORIES.	680601
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COBOL REFERENCE MANUAL PUBLICATION NUMBER 60191200	690901
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS FORTRAN REFERENCE MANUAL PUBLICATION NUMBER 60174900	691103
IBM SYSTEM/360 OPERATING SYSTEM FULL AMERICAN NATIONAL STANDARD COBOL PROGRAMMERS GUIDE NUMBERS 360S-CB-545	700602
IBM SYSTEM/360 OPERATING SYSTEM FORTRAN IV (G AND H) PROGRAMMERS GUIDE PROGRAM NUMBERS 360S-FO-500 360S-FO-520	700603
IBM SYSTEM/360 OPERATING SYSTEM PL/I(F) PROGRAMMERS GUIDE PROGRAM NUMBER 360S-NL-511	710101
UNIVAC 1100 SERIES OPERATING SYSTEM	710102

DATA-PROCESSING
CONTINUATION

CONVERSION OF DECISION TABLES TO COMPUTER PROGRAMS BY RULE MASK TECHNIQUES	661101
PARSING OF DECISION TABLES	670803
CONVERSION OF LIMITED-ENTRY DECISION TABLES TO OPTIMAL COMPUTER PROGRAMS II: MINIMUM STORAGE REQUIREMENT	671102
COMPUTERIZING A GOVERNMENT DATA SYSTEM: A MANAGEMENT OVERVIEW OF THE STEPS REQUIRED AND THE TIME NEEDED.	671202
SIMULATION WITH DECISION TABLES	680101
USER'S MANUAL NMCS INFORMATION PROCESSING SYSTEM (NIPS) (IBM 1410/7010)	680102
BIBLIOGRAPHY ON INFORMATION SCIENCES. VOLUME I	680201
STUDY OF MACHINE-AIDED POST EDITING	680202
NMCS INFORMATION PROCESSING SYSTEM IBM 1410/7010 GENERAL PURPOSE COMPONENTS MULTIFILE OUTPUT AND FILE GENERATION	680302
S/360 ASSEMBLER/COMPILER LISTINGS LIBRARY MAINTENANCE, NO. 360D-00.0.015	681101
DOCUMENTATION STANDARDS	690103
VIEWS ON COMPUTER PROGRAM DOCUMENTATION AND AUTOMATION	701107
MICROFICHE, MICROFILM, AND RELATED EQUIPMENT	711001
DATA COMPRESSION IN LARGE BUSINESS-ORIENTED FILES	711002
ON-LINE TEXT EDITING: A SURVEY	720604

DATA-PROCESSING-TERMINALS

CONTROL DATA 6000 SERIES COMPUTER SYSTEMS REFERENCE MANUAL NUMBER 60100000	650701
WYLBUR	700903
ARPA NETWORK SERIES I.INTRODUCTION TO THE ARPA NETWORK AT RAND AND TO THE RAND VIDEO GRAPHICS SYSTEM	710901

DATA-STORAGE

EVOLUTIONARY SYSTEM FOR DATA PROCESSING, VOLUMES I-IV.	000102
DATA, DOCUMENTATION, AND DECISION TABLES	660101
INFORMATION SCIENCES: SOME RESEARCH DIRECTIVES	660401
SLIP - SOURCE LIBRARY INQUIRY PROGRAM, NO. 360D-00.3.009	660601
CONVERSION OF LIMITED-ENTRY DECISION TABLES TO OPTIMAL COMPUTER PROGRAMS II: MINIMUM STORAGE REQUIREMENT	671102
AUTOMATION IN LINGUISTICS	680103
S/360 ASSEMBLER/COMPILER LISTINGS LIBRARY MAINTENANCE, NO. 360D-00.0.015	681101
IBM SYSTEM/360 OPERATING SYSTEM LINKAGE EDITOR AND LOADER PROGRAM NUMBERS 360S-ED-510 360S-ED-521 360S-LD-547	691102
ON THE CONVERSION OF DECISION TABLES TO COMPUTER PROGRAMS	700601
MICROFICHE STORAGE AND RETRIEVAL SYSTEMS STUDY:	700801
DATA COMPRESSION IN LARGE BUSINESS-ORIENTED FILES	711002
LIBRARIAN	720601

DATA-TRANSMISSION

DATA, DOCUMENTATION, AND DECISION TABLES	660101
CONVERSION OF LIMITED-ENTRY DECISION TABLES TO OPTIMAL COMPUTER PROGRAMS II: MINIMUM STORAGE REQUIREMENT	671102
IBM SYSTEM/360 OPERATING SYSTEM LINKAGE EDITOR AND LOADER PROGRAM NUMBERS 360S-ED-510 360S-ED-521 360S-LD-547	691102
AUTOMATIC SELECTIVE DOCUMENTATION SERVICES	710302
MICROFICHE, MICROFILM, AND RELATED EQUIPMENT	711001

DECISION-TABLES

DECISION-TABLES
CONTINUATION

DESIGN

DESIGN
CONTINUATION

DESIGN
CONTINUATION

DIAGNOSTIC-ROUTINES

DIAGNOSTIC-ROUTINES
CONTINUATION

DOCUMENTATION SOFTWARE	670801
DEVELOPMENT OF A COMPUTER PROGRAM FOR GENERATING TROUBLE-SHOOTING DECISION TREES	670901
COBOL FLOWCHARTING	680301
AMBIGUITY IN LIMITED ENTRY DECISION TABLES	681001
THE INTERPRETATION OF LIMITED ENTRY DECISION TABLE FORMAT AND RELATIONSHIPS AMONG CONDITIONS	691101
A GUIDE FOR SOFTWARE DOCUMENTATION	700001
ON THE CONVERSION OF DECISION TABLES TO COMPUTER PROGRAMS	700601
RESEARCH IN COMPUTER-ASSISTED DOCUMENTATION OF COMPUTER PROGRAMS	701202
PERSPECTIVE ON FLOWCHART PACKAGES	710301

DICTIONARIES

CONTROL DATA 6000 SERIES COMPUTER SYSTEMS REFERENCE MANUAL NUMBER 60100000	650701
DATA, DOCUMENTATION, AND DECISION TABLES	660101
UNIVAC 1108 COBOL UP-4048	670101
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COBOL REFERENCE MANUAL PUBLICATION NUMBER 60191200	690901
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COMPASS REFERENCE MANUAL PUBLICATION NUMBER 60190900	691001
IBM SYSTEM/360 OPERATING SYSTEM LINKAGE EDITOR AND LOADER PROGRAM NUMBERS 360S-ED-510 360S-ED-521 360S-LD-547	691102
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS FORTRAN REFERENCE MANUAL PUBLICATION NUMBER 60174900	691103
IBM SYSTEM/360 OPERATING SYSTEM FULL AMERICAN NATIONAL STANDARD COBOL PROGRAMMERS GUIDE NUMBERS 360S-CB-545	700602
IBM SYSTEM/360 OPERATING SYSTEM FORTRAN IV	

DICTIONARIES
CONTINUATION

(G AND H) PROGRAMMERS GUIDE PROGRAM NUMBERS 360S-FO-500 360S-FO-520	700603
IBM SYSTEM/360 OPERATING SYSTEM ASSEMBLER F PROGRAMMERS GUIDE PROGRAM NUMBER 360L-AS-037	701203
IBM SYSTEM/360 OPERATING SYSTEM PL/I(F) PROGRAMMERS GUIDE PROGRAM NUMBER 360S-NL-511	710101
UNIVAC 1100 SERIES OPERATING SYSTEM	710102

DIRECTORIES

GOOD DECISION TABLES AND THEIR USES	680901

DOCUMENTATION-AID

EVOLUTIONARY SYSTEM FOR DATA PROCESSING, VOLUMES I-IV.	000102
COMCHART	720605
SUPEREF	720606
FORTRAN VARIABLE NAME DOCUMENTER	720607
FORFLO	720608
FLOWGEN/F	720609
DYNACHART	720610
QUICK-DRAW	720611
AUTOFLOW	720612
AUTOCHART	720613
AUTODIAGRAMMER II	720614
AUTODOC	720615
EZFLOW	720616

DOCUMENTATION-COST

EVOLUTIONARY SYSTEM FOR DATA PROCESSING, VOLUMES I-IV.	000102
COST ADVANTAGES OF AN INTEGRATED DOCUMENTATION APPROACH	701113

EDITING

EDITING
CONTINUATION

EXECUTIVE-ROUTINES

FLOWCHARTS

FLOWCHARTS
CONTINUATION

FLOWCHARTS
CONTINUATION

FORMS

A-16

INPUT-OUTPUT-ROUTINES
CONTINUATION

THE GROWTH OF COMPUTER GRAPHICS AT BELL LABORATORIES.	680601
A TECHNIQUE FOR COMPUTER FLOWCHART GENERATION.	680801
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COBOL REFERENCE MANUAL PUBLICATION NUMBER 60191200	690901
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COMPASS REFERENCE MANUAL PUBLICATION NUMBER 60190900	691001
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS FORTRAN REFERENCE MANUAL PUBLICATION NUMBER 60174900	691103
A GUIDE FOR SOFTWARE DOCUMENTATION	700001
IBM SYSTEM/360 OPERATING SYSTEM FULL AMERICAN NATIONAL STANDARD COBOL PROGRAMMERS GUIDE NUMBERS 360S-CB-545	700602
IBM SYSTEM/360 OPERATING SYSTEM FORTRAN IV (G AND H) PROGRAMMERS GUIDE PROGRAM NUMBERS 360S-FO-500 360S-FO-520	700603
THE DESIGN OF VISUAL DISPLAYS	700901
IBM SYSTEM/360 OPERATING SYSTEM ASSEMBLER F PROGRAMMERS GUIDE PROGRAM NUMBER 360L-AS-037	701203
IBM SYSTEM/360 OPERATING SYSTEM PL/I(P) PROGRAMMERS GUIDE PROGRAM NUMBER 360S-NL-511	710101
UNIVAC 1100 SERIES OPERATING SYSTEM	710102

LOADER-ROUTINES

CONTROL DATA 6000 SERIES COMPUTER SYSTEMS REFERENCE MANUAL NUMBER 60100000	650701
UNIVAC 1108 COBOL UP-4048	670101
CONTROL SECTION CROSS-REFERENCE AND LINK EDIT EDITOR	681003
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COBOL REFERENCE MANUAL PUBLICATION NUMBER 60191200	690901

LOADER-ROUTINES
CONTINUATION

CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COMPASS REFERENCE MANUAL PUBLICATION NUMBER 60190900	691001
IBM SYSTEM/360 OPERATING SYSTEM LINKAGE EDITOR AND LOADER PROGRAM NUMBERS 360S-ED-510 360S-ED-521 360S-LD-547	691102
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS FORTRAN REFERENCE MANUAL PUBLICATION NUMBER 60174900	691103
IBM SYSTEM/360 OPERATING SYSTEM FULL AMERICAN NATIONAL STANDARD COBOL PROGRAMMERS GUIDE NUMBERS 360S-CB-545	700602
IBM SYSTEM/360 OPERATING SYSTEM FORTRAN IV (G AND H) PROGRAMMERS GUIDE PROGRAM NUMBERS 360S-FO-500 360S-FO-520	700603
IBM SYSTEM/360 OPERATING SYSTEM ASSEMBLER F PROGRAMMERS GUIDE PROGRAM NUMBER 360L-AS-037	701203
IBM SYSTEM/360 OPERATING SYSTEM PL/I(P) PROGRAMMERS GUIDE PROGRAM NUMBER 360S-NL-511	710101
UNIVAC 1100 SERIES OPERATING SYSTEM	710102

MAGNETIC-STORAGE

CONTROL DATA 6000 SERIES COMPUTER SYSTEMS REFERENCE MANUAL NUMBER 60100000	650701
SLIP - SOURCE LIBRARY INQUIRY PROGRAM, NO. 360D-00.3.009	660601
S/360 ASSEMBLER/COMPILER LISTINGS LIBRARY MAINTENANCE, NO. 3600-00.0.015	681101

MAINTENANCE

EVOLUTIONARY SYSTEM FOR DATA PROCESSING, VOLUMES I-IV.	000102
SLIP - SOURCE LIBRARY INQUIRY PROGRAM, NO. 360D-00.3.009	660601
A SYSTEM OF PROGRAM DOCUMENTATION	660901
AUTODOC: COMPUTER-BASED ASSISTANCE FOR DOCUMENT PRODUCTION	670001

MAINTENANCE
CONTINUATION

GOOD DECISION TABLES AND THEIR USES	680901
S/360 ASSEMBLER/COMPILER LISTINGS LIBRARY MAINTENANCE, NO. 360D-00.0.015	681101
A CONVERSATIONAL CONTEXT-DIRECTED EDITOR	690301
DOCUMENTATION STANDARDS	700002
CLUT - CARD LIBRARY ON TAPE	700604
WYLBUR	700903
AUTOMATED ENGINEERING DESIGN AN APPROACH TO AUTOMATED DOCUMENTATION	701108
ED PROCESSOR - UNIVAC 1100 SERIES OPERATING SYSTEM PROGRAMMER REFERENCE	710103
EDIT - KRONOS TEXT EDITOR (EDIT) REFERENCE MANUAL	710104
TECO - TEXT EDITOR AND CORRECTOR - REFERENCE MANUAL	710603
DATA COMPRESSION IN LARGE BUSINESS-ORIENTED FILES	711002
ACHIEVING PROPER PROGRAM DOCUMENTATION	711101
CFMS - CARD FILE MAINTENANCE SYSTEM	720101
PLUS D/A, A DIRECT ACCESS SOFTWARE SECURITY AND CONTROL SYSTEM	720201
PANVALET - THE DIRECT ACCESS LIBRARY SYSTEM	720502
LIBRARIAN	720601
PROGRAM/MANAGE, A PROGRAM TO MAINTAIN SOURCE PROGRAM	720602
DATA MANIPULATION	720603
ON-LINE TEXT EDITING: A SURVEY	720604

MANUALS

CONTROL DATA 6000 SERIES COMPUTER SYSTEMS REFERENCE MANUAL NUMBER 60100000	650701
PROGRAMMING DOCUMENTATION	651001

MANUALS
CONTINUATION

PROGRAM DOCUMENTATION STANDARDS	660403
A SYSTEM OF PROGRAM DOCUMENTATION	660901
UNIVAC 1108 COBOL UP-4048	670101
THE JOSS PRIMER	670804
USER'S MANUAL NMCS INFORMATION PROCESSING SYSTEM (NIPS) (IBM 1410/7010)	680102
NMCS INFORMATION PROCESSING SYSTEM IBM 1410/7010 GENERAL PURPOSE COMPONENTS MULTIFILE OUTPUT AND FILE GENERATION	680302
DOCUMENTATION STANDARDS	690103
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COBOL REFERENCE MANUAL PUBLICATION NUMBER 60191200	690901
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COMPASS REFERENCE MANUAL PUBLICATION NUMBER 60190900	691001
IBM SYSTEM/360 OPERATING SYSTEM LINKAGE EDITOR AND LOADER PROGRAM NUMBERS 360S-ED-510 360S-ED-521 360S-LD-547	691102
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS FORTRAN REFERENCE MANUAL PUBLICATION NUMBER 60174900	691103
MILITARY STANDARDS	700003
RUNOFF: A PROGRAM FOR THE PREPARATION OF DOCUMENTS	700301
IBM SYSTEM/360 OPERATING SYSTEM FULL AMERICAN NATIONAL STANDARD COBOL PROGRAMMERS GUIDE NUMBERS 360S-CB-545	700602
IBM SYSTEM/360 OPERATING SYSTEM FORTRAN IV (G AND H) PROGRAMMERS GUIDE PROGRAM NUMBERS 360S-FO-500 360S-FO-520	700603
COSMIC: PROGRAM DOCUMENTATION EXPERIENCE	701111
IBM SYSTEM/360 OPERATING SYSTEM ASSEMBLER F PROGRAMMERS GUIDE PROGRAM NUMBER 360L-AS-037	701203
IBM SYSTEM/360 OPERATING SYSTEM PL/I(P)	

PLOTTING

AUTOCHART NEW METHOD CUTS MANUFACTURING.	640501
AUTOMATED PLOTTING OF FLOWCHARTS ON A SMALL COMPUTER	650101
THE GROWTH OF COMPUTER GRAPHICS AT BELL LABORATORIES.	680601
A TECHNIQUE FOR COMPUTER FLOWCHART GENERATION.	680801

PRINTING

AUTOMATIC PREPARATION OF FLOWCHART LISTING.	580101
COMPUTER-DRAWN FLOWCHARTS.	630901
PROGRAM ENABLES IBM 7070/7074 TO DRAW FLOWCHARTS.	631001
A FLOW ANALYSIS ALGORITHM	641001
FLOWTRACE, A PROGRAM FOR FLOWCHARTING PROGRAMS.	661201
A PROGRAM FOR THE CONSTRUCTION OF PERT FLOWCHARTS	671101
AN INTERIM PROGRESS REPORT OF COMPUTER OUTPUT MICROFILM ACTIVITIES AND EXPERIENCES	700701
WYLBUR	700903
MICROFICHE, MICROFILM, AND RELATED EQUIPMENT	711001

PROGRAM-CONVERSION

COST ADVANTAGES OF AN INTEGRATED DOCUMENTATION APPROACH	701113

PROGRAM-PREPROCESSOR

THE INTEGRATION OF SYSTEM SPECIFICATIONS AND PROGRAM CODING. A REPORT ON EXPERIENCE OF THE MEDLARS II PROJECT.	701114

PROGRAMMING-LANGUAGES

A PROGRAM TO DRAW MULTILEVEL FLOWCHARTS	590001
PROGRAMMING DECISION TABLES IN FORTRAN,	

PROGRAMMING-LANGUAGES CONTINUATION

COBOL OR ALGOL	660102
INFORMATION SCIENCES: SOME RESEARCH DIRECTIVES	660401
UNIVAC 1108 COBOL UP-4048	670101
COMPUTER IN INFORMATION SCIENCES. VOLUME III	681002
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COBOL REFERENCE MANUAL PUBLICATION NUMBER 60191200	690901
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COMPASS REFERENCE MANUAL PUBLICATION NUMBER 60190900	691001
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS FORTRAN REFERENCE MANUAL PUBLICATION NUMBER 60174900	691103
IBM SYSTEM/360 OPERATING SYSTEM FULL AMERICAN NATIONAL STANDARD COBOL PROGRAMMERS GUIDE NUMBERS 360S-CB-545	700602
IBM SYSTEM/360 OPERATING SYSTEM FORTRAN IV (G AND H) PROGRAMMERS GUIDE PROGRAM NUMBERS 360S-FO-520 360S-FO-520	700603
IBM SYSTEM/360 OPERATING SYSTEM ASSEMBLER F PROGRAMMERS GUIDE PROGRAM NUMBER 360L-AS-037	701203
IBM SYSTEM/360 OPERATING SYSTEM PL/I(F) PROGRAMMERS GUIDE PROGRAM NUMBER 360S-NL-511	710101
A SURVEY OF DATA STRUCTURES FOR COMPUTER GRAPHICS SYSTEMS	710303

PROGRAMMING-TECHNIQUES

TABLES, FLOW CHARTS, AND PROGRAM LOGIC	620901
AUTOMATED PLOTTING OF FLOWCHARTS ON A SMALL COMPUTER	650101
USE OF DECISION TABLES IN COMPUTER PROGRAMMING	650102
CONVERSION OF LIMITED ENTRY DECISION TABLES TO COMPUTER PROGRAMS	651101

PROGRAMMING-TECHNIQUES
CONTINUATION

PROGRAMMING-TECHNIQUES
CONTINUATION

PROGRAMS

SOFTWARE-COMPANIES
CONTINUATION

A LIBRARY OF DATA COLLECTION AND MANIPULATION SUBROUTINES 710902

CFMS - CARD FILE MAINTENANCE SYSTEM 720101

PLUS D/A, A DIRECT ACCESS SOFTWARE SECURITY AND CONTROL SYSTEM 720201

PANVALET - THE DIRECT ACCESS LIBRARY SYSTEM 720502

LIBRARIAN 720601

PROGRAM/MANAGE, A PROGRAM TO MAINTAIN SOURCE PROGRAM 720602

DATA MANIPULATION 720603

SOUND-REPRODUCTION

TALKING DOWN A PROGRAM 720501

SOURCE-PROGRAMS

PROGRAMMING DECISION TABLES IN FORTRAN, COBOL OR ALGOL 660102

SLIP - SOURCE LIBRARY INQUIRY PROGRAM, NO. 3600-00.3.039 660601

S/360 ASSEMBLER/COMPILER LISTINGS LIBRARY MAINTENANCE, NO. 3600-00.0.015 681101

A CONVERSATIONAL CONTEXT-DIRECTED EDITOR 690301

CLOT - CARD LIBRARY ON TAPE 700604

WYLBUR 700903

AUTOMATIC PROGRAM ANNOTATION (AUTONOTE) 701104

BELLFLOW AUTOMATIC FLOWCHARTING SYSTEM 701105

AN AUTOMATED SYSTEM FOR GENERATING PROGRAM DOCUMENTATION 701106

AUTOMATED DOCUMENTATION OF AN ASSEMBLY PROGRAM 701110

ED PROCESSOR - UNIVAC 1100 SERIES OPERATING SYSTEM PROGRAMMER REFERENCE 710103

EDIT - KRONOS TEXT EDITOR (EDIT) REFERENCE

SOURCE-PROGRAMS
CONTINUATION

MANUAL 710104

TECO - TEXT EDITOR AND CORRECTOR - REFERENCE MANUAL 710603

CFMS - CARD FILE MAINTENANCE SYSTEM 720101

PLUS D/A, A DIRECT ACCESS SOFTWARE SECURITY AND CONTROL SYSTEM 720201

PANVALET - THE DIRECT ACCESS LIBRARY SYSTEM 720502

LIBRARIAN 720601

PROGRAM/MANAGE, A PROGRAM TO MAINTAIN SOURCE PROGRAM 720602

DATA MANIPULATION 720603

ON-LINE TEXT EDITING: A SURVEY 720604

STANDARDS

ON THE FEASIBILITY OF FORMAL CERTIFICATION 000103

FLOWCHARTS 000104

A PROGRAM TO DRAW MULTILEVEL FLOWCHARTS 590001

PROGRAMMING DECISION TABLES IN FORTRAN, COBOL OR ALGOL 660102

PROGRAM DOCUMENTATION STANDARDS 660403

A SYSTEM OF PROGRAM DOCUMENTATION 660901

A PROGRAM FOR THE CONSTRUCTION OF PERT FLOWCHARTS 671101

USER'S MANUAL NMCS INFORMATION PROCESSING SYSTEM (NIPS) (IBM 1410/7010) 680102

NMCS INFORMATION PROCESSING SYSTEM IBM 1410/7010 GENERAL PURPOSE COMPONENTS MULTIFILE OUTPUT AND FILE GENERATION 680302

AN INTRODUCTION TO DECISION TABLES 680701

DOCUMENTATION STANDARDS 690103

ON-LINE MANAGEMENT SYSTEMS INFORMATION 690401

TECHNICAL-WRITING

INDEX PREPARATION FOR PUBLICATIONS INDUSTRY — 680902

TEXT

STUDY OF MACHINE-AIDED POST EDITING — 680202

SYSTEM/360 ADMINISTRATIVE TERMINAL SYSTEM--OS — 680303

INDEX PREPARATION FOR PUBLICATIONS INDUSTRY — 680902

RUNOFF: A PROGRAM FOR THE PREPARATION OF DOCUMENTS — 700301

WYLBUR — 700903

AUTOFLOW ENHANCEMENTS OF DOCUMENTATION AND MAINTENANCE OF SCIENTIFIC APPLICATIONS — 701101

AN AUTOMATED SYSTEM FOR GENERATING PROGRAM DOCUMENTATION — 701106

PROGRAM AUTOMATED DOCUMENTATION METHODS — 701109

AUTOMATIC EDITING OF MANUALS — 701112

ED PROCESSOR - UNIVAC 1100 SERIES OPERATING SYSTEM PROGRAMMER REFERENCE — 710103

EDIT - KRONOS TEXT EDITOR (EDIT) REFERENCE MANUAL — 710104

KRONOS TEXT EDITOR (EDIT) — 710105

XEROX TEXT LANGUAGE AND OPERATIONS REFERENCE MANUAL — 720503

ON-LINE TEXT EDITING: A SURVEY — 720604

FRESS USER'S GUIDE — 720617

NETWORK INFORMATION CENTER USER GUIDE — 720701

ASTROCOMP — 720702

TIME-SHARING

INFORMATION SCIENCES: SOME RESEARCH DIRECTIVES — 660401

THE JOSS PRIMER — 670804

TIME-SHARING
CONTINUATION

COMPUTER IN INFORMATION SCIENCES. VOLUME III — 681002

RUNOFF: A PROGRAM FOR THE PREPARATION OF DOCUMENTS — 700301

PROGRAM AUTOMATED DOCUMENTATION METHODS — 701109

EDIT - KRONOS TEXT EDITOR (EDIT) REFERENCE MANUAL — 710104

000101 COMMUNICATIONS COMPUTER PROGRAMMING CENTER
AUTOMATED WEATHER NETWORK
AUTOMATED WEATHER DIVISION, CCPC, TINKER AIR FORCE BASE,
THIS PAPER DESCRIBES THE AUTOMATED WEATHER NETWORK AND MENTIONS SOME SUBROUTINES AND INPUT-OUTPUT HANDLING METHODS USED IN THE PROJECT. THIS IS A VERY SPECIALIZED PAPER OF LITTLE GENERAL INTEREST.

000102 CONKLIN, C.F. MEADOW, C.T.
EVOLUTIONARY SYSTEM FOR DATA PROCESSING, VOLUMES I-IV.
IBM CENTER FOR EXPLORATORY STUDIES, ROCKVILLE, MD, 1968
ESDP IS A PROPOSED SYSTEM WHOSE PURPOSE IS TO ACQUIRE,STORE, RETRIEVE,PUBLISH,AND DISSEMINATE ALL DOCUMENTATION, EXCLUSIVE OF GRAPHICS FOR A LARGE COMPUTER PROGRAMMING ACTIVITY
VOLUME I. SYSTEM DESCRIPTION.
VOLUME II. CONTROL AND USE OF THE SYSTEM
VOLUME III THE CAINT EXECUTIVE LANGUAGE & INSTR GENERATOR
VOLUME IV. PROGRAMMING SPECIFICATIONS

000103 KEIRSTEAD, R.F. PARKER, D.B.
ON THE FEASIBILITY OF FORMAL CERTIFICATION
STANFORD RESEARCH INSTITUTE, MENLO PARK, CALIFORNIA
THIS PAPER CONSIDERS THE FEASIBILITY OF ESTABLISHING CERTIFICATION AND VALIDATION SERVICES TO ASCERTAIN THE RELIABILITY OF SOFTWARE PRODUCTS.

000104 CHAPIN, N.
FLOWCHARTS
AUERBACH PUBLISHERS, PRINCETON, 1971
A COMPREHENSIVE STUDY OF HOW GRAPHIC TECHNIQUES CAN BE USED TO EXPLAIN AND INTERPRET THE COMPLICATED WORKINGS OF A COMPUTER. THE BOOK EXPLAINS HOW TO APPLY FLOWCHARTS TO ALL PHASES OF DATA PREPARATION AND PROCESSING.

580101 SCOTT, A. E.
AUTOMATIC PREPARATION OF FLOWCHART LISTING.
J. ACM 5, 1(JAN,1958), 57-66
COVERS INITIAL MEANS OF AUTOMATING FLOWCHART PRODUCTION, A REASON FOR FLOWCHART PRODUCTION IS TO MAKE IT UNNECESSARY TO PUBLISH DETAILED BLOCK DIAGRAMS. AS A DEBUGGING TOOL, IT HAS GREAT VALUE.

590001 HAIBT, L.M.
A PROGRAM TO DRAW MULTILEVEL FLOWCHARTS
AFIPS WJCC PROC. (1959), 131-137.
GENERALLY COVERS THE VARIOUS TYPES OF FLOWCHARTING.

610601 CANTRELL, H.N. KING, F.E.
LOGIC STRUCTURE TABLES
COMM. ACM 4(JUNE 1961),272-275
DESCRIBES SET OF RULES FOR WRITING AND USING LOGIC TABLES. LOGIC TABLES ARE TWO-DIMENSIONAL ENABLING ONE TO FULLY EXPRESS BOTH SEQUENTIAL AND PARALLEL ASPECTS. THEY CAN BE COMPILED DIRECTLY INTO PROGRAM ELIMINATING NEED FOR FLOWCHARTING AND HAND CODING.

620901 MONTALBANO, M.
TABLES, FLOW CHARTS, AND PROGRAM LOGIC
IBM SYSTEMS J. 1,9(SEPT 1962),51-63.
DECISION TABLES ARE INTRODUCED WITH REFERENCE TO BUSINESS DATA PROCESSING. COMPLETENESS AND CONSISTENCY VERIFICATION CHECKS ARE MADE. CONVERSION OF TABLES TO PROGRAMS IS CONSIDERED AND A TECHNIQUE OF OBTAINING A PROGRAM WHICH MINIMIZES BRANCHING REQUIREMENTS WITH RESPECT TO MEMORY AND EXECUTION TIME. DEBUGGING AND MODIFICATION ARE ALSO DISCUSSED.

630901 KNUTH, H. E.
COMPUTER-DRAWN FLOWCHARTS.
COMM. ACM 6, 9(SEPT,1963), 555-563
SIMPLE SYSTEM FOR EFFECTIVE COMMUNICATION IS PRESENTED. SUPPOSEDLY HELPS MEET FOR BETTER DOCUMENTATION OF PROGRAMS IN SIMPLE FORMAT, COMPUTER PREPARES FLOWCHARTS AND OTHER CROSS-REFERENCE LISTING FROM THIS INPUT.

631001 IBM CORPORATION
PROGRAM ENABLES IBM 7070/7074 TO DRAW FLOWCHARTS.
COMPUTERS AND AUTOMATION 12, 10(OCT.1963), 40.
A GENERAL SYSTEM OF FLOWCHARTING. THE COMPUTER PRINTS OUT A FLOWCHART.

640301 SHMIDT, D.T. KAVANAGH, T.E.
USING DECISION STRUCTURE TABLES
DATAMATION 10,3(MARCH 1964),48
THIS ARTICLE EMPHASIZES MANUFACTURING APPLICATIONS OF DECISION-TABLES. IT ALSO LISTS REASONS WHY DECISION TABLES PAY OFF.

640501 MERCER, V.
AUTOCHART NEW METHOD CUTS MANUFACTURING.
JOURNAL OF DATA MANAGEMENT 2, 5(MAY,1964), 644-645.
COMMERCIAL PROGRAM PACKAGE CAN BE USED TO DECREASE THE COST OF PRODUCING PROGRAMS.

641001 KRIDER, L.
A FLOW ANALYSIS ALGORITHM
J. ACM 11, 10(OCT. 1964), 429-436.
ALGEBRAIC REPRESENTATION OF THE FLOW OF A COMPUTER PROGRAM IS DESCRIBED. ALGORITHM PRESENTED FOR MANIPULATING THIS REPRESENTATION INTO A FORM FROM WHICH A FLOWCHART CAN BE DRAWN. ALSO, A PROCEDURE FOR FORMING THE ALGEBRAIC REPRESENTATION FROM A COMPILER SOURCE CODE IS GIVEN.

650101 ANDERSON, H. E.
AUTOMATED PLOTTING OF FLOWCHARTS ON A SMALL COMPUTER
COMM. ACM 8, 1(JAN 1965), 38-39.
PROGRAM IS DESCRIBED WHICH ACCEPTS FORTRAN II SOURCE PROGRAM AS INPUT. PRODUCES ANNOTATED FLOWCHARTS AS OUTPUT. UNIQUE IN THAT A DIGITAL INCREMENTAL PLOTTER IS USED AS OUTPUT MEDIUM, AND THAT ITS DESIGNED FOR A SMALL (40K) COMPUTER.

650102 KIRK, H.W.
USE OF DECISION TABLES IN COMPUTER PROGRAMMING
COMM. ACM 8,1(JAN 1965),41-43.

650102 ARTICLE DESCRIBES DECISION TABLE ADVANTAGES. A TECHNIQUE FOR CONVERSION OF TABLES TO COMPUTER PROGRAMS IS PRESENTED; IT IS THE RULE MASK TECHNIQUE. ADVANTAGES GAINED THROUGH UTILIZATION OF THE TECHNIQUE ARE ALSO DESCRIBED.

650701 CONTROL DATA CORPORATION
CONTROL DATA 6000 SERIES COMPUTER SYSTEMS REFERENCE MANUAL NUMBER 60100000
CONTROL DATA CORPORATION
THIS MANUAL DESCRIBES THE SYSTEM CHARACTERISTICS OF THE 6600 SERIES. ALSO DISCUSSED ARE THE CENTRAL MEMORY CENTRAL PROCESSOR PERIPHERAL AND CONTROL PROCESSORS SYSTEM INTERRUPTS AND MANUAL CONTROL OF THE SYSTEM.

651001 SNYDER, R.
PROGRAMMING DOCUMENTATION
DATAMATION 11,10(OCTOBER 1965),44-45
DOCUMENTATION IS A BY-PRODUCT OF A GOOD SYSTEMS JOB, BUT IT IS EVEN MORE IMPORTANT DURING COMPUTER SYSTEM CONVERSION. THIS ARTICLE LISTS WHAT GOES INTO AN EFFECTIVE JOB MANUAL.

651101 POLLACK, S.L.
CONVERSION OF LIMITED ENTRY DECISION TABLES TO COMPUTER PROGRAMS
COMM. ACM 8,11(NOV 1965),677-682
TWO ALGORITHMS ARE PRESENTED FOR CONVERSION OF DECISION TABLES TO COMPUTER PROGRAMS. BOTH ALGORITHMS PINPOINT REDUNDANCIES AND CONTRADICTIONS IN THE TABLES. THE FIRST ALGORITHM MINIMIZES STORAGE SPACE, THE SECOND MINIMIZES RUN TIME.

660101 FISHER, D.L.
DATA, DOCUMENTATION, AND DECISION TABLES
COMM. ACM 9,1(JAN 1966),26-31
IN BUSINESS DATA PROCESSING, IT IS NECESSARY TO BE ABLE TO DEFINE AND DOCUMENT DATA, FILES, PROGRAMS AND DECISION RULES IN A WAY THAT REPRESENTS THEIR CHANGING INFORMATION FORMAT AND THEIR CONTINUOUS INTERACTION. TABLES MAKE THIS POSSIBLE. ARTICLE IS BASICALLY A BUILDUP OF THE VIRTUES OF DECISION TABLES.

660102 VEINOTT, C.G.
PROGRAMMING DECISION TABLES IN FORTRAN, COBOL OR ALGOL
COMM. ACM 9,1(JAN 1966),31-35.
A BROAD-BASED APPROACH FOR PROGRAMMING DECISION TABLES IN FORTRAN OR COBOL IS DEVELOPED. ONLY ONE OR TWO STATEMENTS ARE DEEMED NECESSARY WITH INPUT IN A STANDARDIZED FORM. ACTUAL PROGRAMMING TECHNIQUES ARE OFFERED. THE TECHNIQUES ARE ALSO EXTENDED TO ALGOL.

660201 MARTIN, W. W.
FLOWCHARTING: SHORTHAND, ANALYSIS AND MODEL.
JOURNAL OF SYSTEM MANAGEMENT 17, 2(MARCH-APRIL, 1966), 14-22

660301 SHOBER, J.A.
DECISION TABLES FOR BERRER MANAGEMENT SYSTEMS
SYSTEMS AND PROCEDURES J. 17,2(MARCH-APRIL),28-32.

660301 THIS ARTICLE BEGINS WITH A LIST OF BENEFITS GAINED BY DECISION TABLE UTILIZATION. SOME EXAMPLES OF USAGE ARE PRESENTED. DECISION TABLES ARE DIRECTLY COMPARED TO FLOWCHARTS.

660401 SWANSON, R.W.
INFORMATION SCIENCES: SOME RESEARCH DIRECTIVES
NTIS-AD631535
A STUDY OF THE PROCESSES INVOLVED IN IDENTIFYING, CLASSIFYING, REPRESENTING, STORING, MANIPULATING, AND PRESENTING DATA.

660402 OPERATIONS DIVISION STRATEGIC SYSTEMS DEPARTMENT
AUTOMATIC DATA PROCESSING SUPPORT FOR FLAG PLOT
DDC-AD815567
THIS REPORT PROPOSES PRELIMINARY DESIGNS FOR THE FLAG PLOT ADP SYSTEMS. IT OUTLINES BOTH HARDWARE AND SOFTWARE PLANS FOR THE INITIAL AND INTERMEDIATE EFFORTS IN AUTOMATING FLAG PLOT DATA PROCESSING FUNCTIONS.

660403 BELL SYSTEMS PRACTICES
PROGRAM DOCUMENTATION STANDARDS
BELL SYSTEMS PRACTICES, AMERICAN TELEPHONE AND TELEGRAPH CO.
THIS IS A MANUAL WHICH HAS THE STANDARDS FOR THE BELL SYSTEM IT INCLUDES STANDARDS FOR PROGRAM DOCUMENTATION, FLOWCHARTS SYMBOLS AND NOTATIONS. DECISION TABLES ARE INCLUDED AS WELL AS NUMEROUS EXAMPLES.

660601 PENSIERO, W.S.
SLIP - SOURCE LIBRARY INQUIRY PROGRAM, NO. 360D-00.3.009
IBM CORPORATION, NEW YORK, NY, 1966
THIS PROGRAM STORES SYMBOLIC PROGRAM MODULES ON MAG TAPE AND RETRIEVES SELECTED MODULES TO BE LISTED, PUNCHED, OR COPIED TO ANOTHER MAG TAPE 'IN A FORMAT ACCEPTABLE TO THE ASSEMBLER'. IT RUNS ON A S/360-30 OR LARGER, UNDER 8K BPS.

660801 REINWALD, L.T. SOLAND, R.M.
CONVERSION OF LIMITED-ENTRY DECISION TABLES TO OPTIMAL COMPUTER PROGRAMS I: MINIMUM AVERAGE PROCESSING TIME
J. ACM 13,7(JULY 1966),339-358.
BEGINS WITH A BRIEF DESCRIPTION OF DECISION TABLES AND THEN PRESENTS A DISCUSSION OF ALTERNATE EXPRESSIONS FOR THEM IN THE FORM OF SEQUENTIAL TESTING PROCEDURES. AN ALGORITHM IS DEVELOPED FOR CONVERSION OF TABLES TO A PROGRAM WITH MINIMUM AVERAGE PROCESSING TIME.

660901 HOLZER, J.E.
A SYSTEM OF PROGRAM DOCUMENTATION
SYSTEMS AND PROCEDURES J. 17,5(SEPT-OCT 1966),15-17.
ARTICLE RELATES WHY DOCUMENTATION SHOULD BE USED TO ACHIEVE EFFECTIVE COMPUTER UTILIZATION IN AN ORGANIZATION. A DESCRIPTION OF THE FACETS INVOLVED IN PROGRAM DOCUMENTATION IS ALSO PRESENTED. A MANUAL CONTAINING ALL DOCUMENTATION IS SUGGESTED.

661101 KING, P.J.H.
CONVERSION OF DECISION TABLES TO COMPUTER PROGRAMS BY RULE MASK TECHNIQUES

661101 COMM. ACM 9,11(NOV 1966),796-801.
ARTICLE PRESENTS A DESCRIPTION OF THE RULE MASK TECHNIQUE FOR CONVERSION OF DECISION TABLES TO COMPUTER PROGRAMS. THE TECHNIQUE, AS PRESENTED, MAY GIVE UNDESIRABLY LONG RUN TIMES. A MODIFICATION IS PROPOSED TO HELP ALLEVIATE THIS PROBLEM.

661201 SHERMAN, PHILIP M.
FLOWTRACE, A PROGRAM FOR FLOWCHARTING PROGRAMS.
COMM. ACM 9, 12(DEC. 1966), 845-854.
FLOWTRACE PRODUCES FLOWCHARTS OF PROGRAMS WRITTEN IN ALMOST ANY PROGRAMMING LANGUAGE. META LANGUAGE AVAILABLE FOR DESCRIPTION OF SYNTAX OF CONTROL STATEMENTS.

670001 CALLAHAM, M.D. GRACE, G.L.
AUTODOC: COMPUTER-BASED ASSISTANCE FOR DOCUMENT PRODUCTION
PROC. ACM CONF 22 (1967), 177-185
AUTODOC IS A MAN-MACHINE DESIGNED TO PROVIDE AUTOMATED DOCUMENTATION PLUS THIS SYSTEM PERMITS THE ADDED FEATURES OF DOCUMENT PRODUCTION, MAINTENANCE, AND BOOKEEPING.

670101 SPERRY RAND CORPORATION
UNIVAC 1108 COBOL UP-4048
UNIVAC CORPORATION
THIS MANUAL EXPLAINS THE COBOL-LANGUAGE AS IT APPLIES TO THE UNIVAC 1108 COMPUTER.

670401 TUPAC, J.D.
AN APPROACH TO SOFTWARE EVALUATION
NTIS-AD651812
A PROCEDURE FOR SOFTWARE EVALUATION IS INTRODUCED. MAJOR IDEAS ARE WHEN AND HOW A MANAGER SHOULD EVALUATE. TWO MAJOR AREAS OF SOFTWARE EVALUATION ARE FUNCTIONAL CAPABILITIES AND PERFORMANCE.

670402 SALISBURY, A.B. ENSLOW,JR. P.H.
DIAGNOSTIC PROGRAMMING FOR DIGITAL COMPUTERS A BIBLIOGRAPHY
NTIS-AD813831
THIS BIBLIOGRAPHY CONSISTS OF 88 REFERENCES ON DIAGNOSTIC PROGRAMMING FOR DIGITAL COMPUTERS. EACH ENTRY INCLUDES AUTHOR(S),TITLE,SOURCE,AN ABSTRACT (NORMALLY THE AUTHOR'S) AND USUALLY SOME COMMENTS PERTAINING TO ITEMS OF SPECIAL INTEREST OR ADDITIONAL INFORMATION TO HELP THE READER DECIDE IF THE DOCUMENT WOULD BE OF INTEREST TO HIM.

670801 COMPRESS, INC.
DOCUMENTATION SOFTWARE
DATAMATION 13, 8(AUGUST 1967), 113-125
DOPIC (DOCUMENTATION OF PROGRAMS IN CORE) GENERATES MACHINE PRODUCED FLOWCHARTS, CROSS-REFERENCE LIST AND CHART INDEX FOR OBJECT PROGRAMS.

670802 KING, P.J.H.
DECISION TABLES
COMPUTER J. 10,8(AUG 1967),135-142.
THE ARTICLE DESCRIBES THE BASIC FEATURES OF DECISION TABLE DEVELOPMENT, AND IT ILLUSTRATES APPLICATIONS IN VARIOUS FIELDS. SOME PROGRAMMING TECHNIQUES ARE PRESENTED. AN

670802 EXTENSIVE DECISION TABLE BIBLIOGRAPHY IS ALSO PRESENTED WITH EXPLANATION OF THE RELATIVE MERITS OF EACH.

670803 CHAPIN, N.
PARSING OF DECISION TABLES
COMM. ACM 10,8(AUG 1967),507-510.
DEALS WITH REDUCTION OF DECISION TABLES BY SEVERAL TECHNIQUES. TECHNIQUES CONSIDERED ARE BY PARSING WITH REGARD TO VERTICAL AND HORIZONTAL DATE STRUCTURES, JOB IDENTITY, HARDWARE AND JOB PRIORITY, AND CONTEXT RELATIONSHIPS. THE PARSING RELIES ON CONVENTIONS FOR DECISION TABLE LINKAGE.

670804 MARKS, S.L. ARMERDING, G.W.
THE JOSS PRIMER
NTIS-AD659734
AN INTRODUCTION TO THE JOSS TIMESHARING SYSTEM. IT IS AN ON-LINE, TIME-SHARED COMPUTING SERVICE DEVELOPED AT THE RAND CORPORATION TO PROVIDE THE SOLUTION OF NUMERICAL PROBLEMS.

670901 HANNOM, T.J. AZZARI, A.I. BROOKS, A.I.
HUNTER, J.R. STEINBERG, L. CHUBB, G.P.
DEVELOPMENT OF A COMPUTER PROGRAM FOR GENERATING TROUBLE-SHOOTING DECISION TREES
NTIS-AD664603
ESTABLISHES ALGORITHM FOR GENERATING A DECISION LOGIC-STRUCTURE OR TREE DIAGRAM FORM WHICH PROCEDURAL TROUBLE SHOOTING INSTRUCTIONS COULD BE PREPARED.

671101 HOWDEN, W. E.
A PROGRAM FOR THE CONSTRUCTION OF PERT FLOWCHARTS
COMPUTER JOURNAL 10, 11(NOV. 1967), 278-281
APPLICATIONS OF CPM AND PERT OFTEN INVOLVE FLOWCHARTS AND NETWORKS. PERT DIAGRAMS MUST BE EASY TO READ TO BE USEFUL. IN THIS ARTICLE, EVENTS AND ACTIVITIES THAT FORM A SUBNETWORK ARE GROUPED TOGETHER AND PROXIMITY OF ANY TWO NETWORKS REFLECTS THEIR INTERDEPENDENCE.

671102 REINWALD, L.T.
CONVERSION OF LIMITED-ENTRY DECISION TABLES TO OPTIMAL COMPUTER PROGRAMS II: MINIMUM STORAGE REQUIREMENT
J. ACM 14,10(OCT 1967),742-756.
AN ALGORITHM IS DEVELOPED WHICH WILL CONVERT ANY LIMITED ENTRY DECISION TABLE TO A COMPUTER PROGRAM WITH MINIMAL STORAGE REQUIREMENT. IT IS INITIALLY DEVELOPED FOR PROGRAMS WITH A TREE-LIKE FORM AND THEN EXTENDED. THIS ALGORITHM CAN BE COMBINED WITH ONE FOR MINIMUM AVERAGE PROCESSING TIME, RESULTING IN A MINIMUM AVERAGE COST PROGRAM.

671201 COMP. SCI. DEPT. UNIVERSITY OF ILLINOIS
ILLIAC IV QUARTERLY PROGRESS REPORT
NTIS-AD667280
A REPORT ON THE PROGRESS OF THE ILLIAC IV SYSTEM IS LISTED. THIS INCLUDES THE GRAPHICAL DISPLAY SYSTEM.

671202 JONES, M.V.

671202 COMPUTERIZING A GOVERNMENT DATA SYSTEM: A MANAGEMENT OVERVIEW OF THE STEPS REQUIRED AND THE TIME NEEDED.
NTIS-AD692784
THIS PAPER DISCUSSES WHAT MUST BE DONE AND HOW LONG IT WILL TAKE TO CONVERT A MANUAL DATA SYSTEM IN A GOVERNMENT AGENCY TO A COMPUTERIZED SYSTEM.

680101 LUDWIG, H.R.
SIMULATION WITH DECISION TABLES
J. DATA MGMT. 6,1(JAN 1968),20-27.
THE ARTICLE DEALS PRIMARILY WITH A SIMULATION PROCEDURE, USING THE DECISION TABLE STRUCTURE TO GOVERN THE BEHAVIOR OF THE QUEUING ACTION. A SIMPLIFIED ALGORITHM IS PRESENTED TO SHOW SOMEWHAT HOW THE DECISION TABLE STRUCTURE WAS USED.

680102 NATIONAL MILITARY COMMAND SYSTEM SUPPORT CENTER COMPUTER
IBM CORPORATION
USER'S MANUAL NMCS INFORMATION PROCESSING SYSTEM (NIPS) (IBM 1410/7010)
NTIS-AD830717
NMCS INFORMATION PROCESSING SYSTEM (IBM 1410/7010) IS DESCRIBED IN SIX DOCUMENTS (1)SYSTEM DESCRIPTION, (2)OPERATOR'S MANUAL,(3)USER'S MANUAL,(4)ANALYTICAL MANUAL (5)PROGRAM DESCRIPTION, AND (6)DATA BASE DOCUMENTATION

680103 MEL'CHUK, I.A.
AUTOMATION IN LINGUISTICS
NTIS-AD734215
THIS ARTICLE RELATES THE PROBLEMS OF AUTOMATION IN LINGUISTICS. REVIEWS AND EXERTS OF TRANSLATIONS BY EXPERTS ARE DISCUSSED.

680201 DEFENSE DOCUMENTATION CENTER
BIBLIOGRAPHY ON INFORMATION SCIENCES. VOLUME I
DDC-AD829001
THIS BIBLIOGRAPHY IS A COMPILATION OF REFERENCES IN THE AREA OF INFORMATION SCIENCE AND TECHNOLOGY. THE REFERENCES REPRESENT RESEARCH ON THE BASIC ACTIVITIES COMMON TO MOST INFORMATION-ORIENTED MISSIONS.

680202 DEMUTH, N. CONTI, E.
STUDY OF MACHINE-AIDED POST EDITING
DDC-AD828584
THE PRIMARY OBJECTIVE OF THIS REPORT WAS TO DETERMINE THE FEASIBILITY OF MACHINE-AIDED POST EDITING AT FTD. ERROR DETECTION ANALYSIS WAS USED TO DEVELOPE AN EXPERIMENTAL POST-EDITING SYSTEM (XPE).

680301 NATIONAL COMPUTER ANALYSIS, INC.
COBOL FLOWCHARTING
DATAMATION 14, 3(MARCH 1968), 135.
NATIONAL COMPUTER ANALYSIS, INC. PRINTS AND MAILS BACK FLOWCHARTS ARE PRODUCED FOR AND COBOL PROGRAM DECKS.

680302 IBM CORPORATION
NMCS INFORMATION PROCESSING SYSTEM IBM 1410/7010 GENERAL PURPOSE COMPONENTS MULTIFILE OUTPUT AND FILE GENERATION

680302 NTIS-AD-834970
NMCS INFORMATION PROCESSING SYSTEM (IBM 1410/7010) IS DESCRIBED IN SIX DOCUMENTS (1)SYSTEM DESCRIPTION, (2)OPERATOR'S MANUAL,(3)USER'S MANUAL,(4)ANALYTICAL MANUAL (5)PROGRAM DESCRIPTION, AND (6)DATA BASE DOCUMENTATION

680303 IBM CORPORATION
SYSTEM/360 ADMINISTRATIVE TERMINAL SYSTEM--OS
IBM CORPORATION
THIS SYSTEM CONSISTS OF CONTROL AND FUNCTIONAL PROGRAMS THAT PERMIT MANY DIFFERENT TEXT-PROCESSING AND DATA-HANDLING ACTIVITIES TO BE CARRIED ON SIMULTANEOUSLY THROUGH DIFFERENT TERMINALS ATTACHED TO AN IBM SYSTEM/360.

680501 CHAPIN, N.
PROGRAM DOCUMENTATION: THE VALUABLE BURDEN
SOFTWARE AGE, 2(MAY 1968),24-26
ADVANTAGES AND OBJECTIVES OF DOCUMENTATION ARE DISCUSSED. TWELVE MAJOR DOCUMENTATION ELEMENTS ARE LISTED.

680601 NINKE, W. H.
THE GROWTH OF COMPUTER GRAPHICS AT BELL LABORATORIES.
BELL LABORATORY RECORD 46, 6(JUNE,1968), 180
COMPUTER GRAPHICS IS A USEFUL TECHNIQUE FOR COMMUNICATING WITH COMPUTERS. THIS TECHNIQUE IS HELPING MAN MAKE MORE EFFECTIVE USE OF THE MACHINE TO SOLVE A WIDE VARIETY OF PROBLEMS WHILE PROVIDING A RECORD OF PROBLEM SOLUTION.

680701 FERGUS, R.M.
AN INTRODUCTION TO DECISION TABLES
SYSTEMS AND PROCEDURES J. 19,4(JULY-AUG 1968),24-27.
THIS ARTICLE DESCRIBES THE CHARACTERISTICS OF DECISION TABLES, ALSO GIVING GUIDELINES AND EXAMPLES OF TABLE CONSTRUCTION AS INITIAL STEPS TOWARD THE USE OF TABLES TO PERFORM, COMMUNICATE AND DOCUMENT DECISION MAKING.

680801 O'BRIEN, F. BECKWITH, R.C.
A TECHNIQUE FOR COMPUTER FLOWCHART GENERATION.
COMPUTER JOURNAL 11, 8(AUG. 1968), 138-140.
AN APPROACH TO COMPUTER FLOWCHART GENERATION CALLED ICFLOW. PROGRAM IS SIMPLE AND EASY AID WHICH PRODUCE AN UNANNOTATED LINEAR FLOWCHART. THE FLOWCHARTS ANNOTATION HAS BEEN LEFT TO THE PROGRAMMER CONCERNED.

680802 GRINDLEY, C.B.
THE USE OF DECISION TABLES WITHIN SYSTEMATICS
COMPUTER J. 11,8(AUG 1968),129-133.
SYSTEMATICS IS A SET OF TECHNIQUES FOR DESIGNING AND DEVELOPING INFORMATION SYSTEMS. A GENERALIZED DESCRIPTION OF DECISION TABLES IS PRESENTED. THE FORM OF DECISION TABLE USED IN A SYSTEMATICS TECHNIQUE IS DESCRIBED ALONG WITH ADVANTAGES GAINED FROM IT.

680901 FERGUS, R.M.
GOOD DECISION TABLES AND THEIR USES
SYSTEMS AND PROCEDURES J. 19,5(SEPT-OCT 1968),18-21.
SEVEN STEPS ARE GIVEN FOR IMPLEMENTATION OF DECISION TABLE USE WITHIN AN ORGANIZATION. TABLES CAN BE USED AT MANY

680901 LEVELS OF AN ORGANIZATION. TABLES CAN BE ANALYZED TO PROMOTE COMPLETENESS AND ACCURACY AND TO DETECT EXCESSIVE RULES, CONTRADICTIONS.

680902 IBM CORPORATION
INDEX PREPARATION FOR PUBLICATIONS INDUSTRY
IBM CORPORATION #3600-29.4.004
THIS PROGRAM PRODUCES AN INDEX IN DOUBLE-COLUMN FORMAT WHICH IS SUITABLE FOR REPRODUCTION BY PHOTO OFFSET PRINTING. THE PROGRAM WOULD BE OF VALUE TO ANY PUBLICATION DEPARTMENT THAT HAS TO PREPARE INDEXES AS A PART OF ITS PUBLICATIONS.

681001 KING, P.J.H.
AMBIGUITY IN LIMITED ENTRY DECISION TABLES
COMM. ACM 11,10(OCT 1968),680-684.
AUTHOR CLAIMS RULES CONCERNING REDUNDANCY, CONTRADICTION AND COMPLETENESS AS ESTABLISHED IN 1963 ARE UNSATISFACTORY. AMBIGUITY IS CLAIMED TO BE AN IMPORTANT ASPECT IN CHECKING TABLES. PROCEDURE FOR PRODUCING CHECKED OUT TABLES IS PROPOSED. IMPORTANCE OF WELL-DESIGNED DIAGNOSTIC FACILITIES IS EMPHASIZED.

681002 DEFENSE DOCUMENTATION CENTER
COMPUTER IN INFORMATION SCIENCES: VOLUME III
DDC-AD846300
THIS BIBLIOGRAPHY COMPILES REFERENCES DEALING SPECIFICALLY WITH THE ROLE OF COMPUTERS IN INFORMATION SCIENCES.

681003 MALTBY, Q.J.
CONTROL SECTION CROSS-REFERENCE AND LINK EDIT EDITOR
IBM CONTRIBUTED PROGRAM LIBRARY
THESE TWO ASSEMBLER PROGRAMS MAY PROVE USEFUL IN ANALYSIS AND DEBUGGING WORK WITH RESPECT TO LARGE AND COMPLEX PLANNED OVERLAY MODULES.

681101 ROSEN, M.H. LOBACK, L.R.
S/360 ASSEMBLER/COMPILER LISTINGS LIBRARY MAINTENANCE, NO. 360D-00.0.015
IBM CORPORATION, HAWTHORNE, NY, 1968
THIS PROGRAM CREATES AND MAINTAINS A LIBRARY OF CURRENT DOS ASSEMBLER AND/OR COMPILER (FORTRAN, RPG, COBOL, PL/I) OUTPUT LISTINGS ON IBM 2311 OR 2314 DASD, USING A SPACE COMPRESSION TECHNIQUE. IT RUNS ON ANY S/360 UNDER DOS, SUPPORTING DAM (DIRECT ACCESS METHOD), REQUIRING AT LEAST 16K OF STORAGE.

690101 IBM CORPORATION
PROBLEM LANGUAGE ANALYZER (PLAN) USERS' INTRODUCTION
IBM CORPORATION,WHITE PLAINS,N.Y.,1969
THE IBM PROBLEM LANGUAGE ANALYZER(PLAN) WAS DESIGNED TO LOWER THE COST OF DEFINING,IMPLEMENTING,AND USING PROBLEM-ORIENTED LANGUAGES. PLAN IS A SET OF PROGRAMS THAT OPERATES ON THREE LEVELS A SUPPORT LEVEL TO HELP THE PLAN PROGRAMMER PRODUCE NEW MODULES TO GIVE APPLICATION DESIGNERS STANDARD COMMANDS FOR CATALOGING THE SEMANTICS OF A NEW POL TO ACCEPT INPUT STATEMENTS,EXECUTE AND PRODUCE RESULTS

690102 IBM CORPORATION

690102 SYSTEM/360 FLOWCHART USER'S MANUAL
OPERATOR'S MANUAL
IBM CORPORATION,WHITE PLAINS,N.Y.,1969
THE IBM SYSTEM/360 FLOWCHART PRODUCES PROGRAM FLOWCHARTS UNDER THE IBM DOS. THIS DOCUMENTATION AID IS INTENDED TO MINIMIZE THE PLANNING AND EFFORT REQUIRED TO PRODUCE AND MAINTAIN PROGRAM DOCUMENTATION.

690103 GRAY, M. LONDON, K.R.
DOCUMENTATION STANDARDS
BRANDON/SYSTEMS PRESS, PRINCETON, 1969
THIS BOOK GIVES AN OPINIONATED OVERVIEW OF WHAT DOCUMENTATION STANDARDS SHOULD BE. INCLUDED ARE SYSTEMS DEVELOPMENT DOCUMENTATION AND PROGRAM DOCUMENTATION FROM A DATA PROCESSING VIEW-POINT. NUMEROS FORMS AND EXAMPLES ARE SHOWN. FLOWCHARTS, DECISION TABLE AND SYMBOL STANDARDS (NATIONAL) ARE INCLUDED IN THE APPENDICES.

690301 IBM CORPORATION
A CONVERSATIONAL CONTEXT-DIRECTED EDITOR
IBM CAMBRIDGE SCIENTIFIC CENTER, CAMBRIDGE, MA (1969)
THIS ON-LINE EDITOR OPERATES UNDER THE CP-67/CMS AND ON-LINE/OS SYSTEMS. IT ALLOWS EDITING OF FILES LARGER THAN AVAILABEL CORE SIZE ONLY BY BRINGING IN A BLOCK AT A TIME FOR EDITING, WITH NO BACKWARD BLOCK ACCESSION ALLOWED.

690401 REAM, N.J.
ON-LINE MANAGEMENT SYSTEMS INFORMATION
DATAMATION 10,4(APRIL 1969),39
DICUSSES THE NEED FOR PROGRAMMING STANDARDS AND A FORMAT SYSTEM OF DOCUMENTATION WHICH HELPS DETERMINE THE RELATIVE SUCCESS OF THE EFFORT IN COMPLETING THESE STANDARDS.

690901 CONTROL DATA CORPORATION
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COBOL REFERENCE MANUAL PUBLICATION NUMBER 60191200
CONTROL DATA CORPORATION
THIS MANUAL EXPLAINS HOW TO USE VERSION 3.1.6 OF THE SCOPE OPERATING SYSTEM TO EXECUTE PROGRAMS WRITTEN IN VERSION 2.1 OF CDC 6400/6500/6600 COBOL.

691001 CONTROL DATA CORPORATION
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COMPASS REFERENCE MANUAL PUBLICATION NUMBER 60190900
CONTROL DATA CORPORATION
THIS MANUAL DISCUSSES PROGRAM STRUCTURE AND ORGANIZATION, COMPASS LANGUAGE CODING, OPERATION CODES, PSEUDO INSTRUCTIONS, MACROS, MICROS, AND ASSEMBLER INPUT/OUTPUT UNDER THE COMPASS SYSTEM.

691101 KING, P.J.H.
THE INTERPRETATION OF LIMITED ENTRY DECISION TABLE FORMAT AND RELATIONSHIPS AMONG CONDITIONS
COMPUTER J. 12,11(NOV 1969),320-326.
THIS ARTICLE EMPHASIZES THE INTERPRETATION OF BASIC FORMAT. RELATIONSHIPS AMONG CONDITIONS ARE SHOWN. A SEPARATE MATTER FROM BASIC FORMAT. FORMAL DEFINITIONS ARE PROPOSED.

691102 IBM CORPORATION
IBM SYSTEM/360 OPERATING SYSTEM LINKAGE EDITOR AND LOADER
PROGRAM NUMBERS 360S-ED-510 360S-ED-521 360S-LD-547
IBM CORPORATION
THE MANUAL DESCRIBES THE LINKAGE EDITOR AND LOADER WHICH ARE
TWO PROGRAMS THAT PREPARE THE OUTPUT OF THE LANGUAGE
TRANSLATORS FOR EXECUTION.

691103 CONTROL DATA CORPORATION
CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS FORTRAN
REFERENCE MANUAL PUBLICATION NUMBER 60174900
CONTROL DATA CORPORATION
THIS MANUAL EXPLAINS HOW TO USE FORTRAN VERSION 2.3 UNDER
VERSION 3 OF THE SCOPE OPERATING SYSTEM.

700001 WALSH, D.
A GUIDE FOR SOFTWARE DOCUMENTATION
MCGRAW-HILL,NY,NY,1970
CONTAINS 14 CAREFULLY PREPARED MODELS OF DOCUMENTATION WHICH
MUST BE PREPARED FOR COMPUTER SOFTWARE AND APPLICATIONS
PRODUCTS.

700002 GRAY, M.
DOCUMENTATION STANDARDS
BRANDON/SYSTEMS PRESS,INC.,NEW YORK, NEW YORK, 1969.
BOOK WAS WRITTEN TO PROVIDE DP MANAGERS, SUPERVISORS AND
ANALYSIS A PRACTICAL GUIDE FOR DESIGN AND IMPLEMENTATION
OF STANDARD DOCUMENTATION SYSTEM. AUTHOR SURVEYS THE TYPES
OF DOCUMENTATION. THE CONTROL AND MAINTENANCE OF SAID
DOCUMENTATION, AND, IN APPENDIX FORM, HAS AN INDEXED
GLOSSARY OF FORMS USED, FLOWCHARTING STANDARDS, AND
DECISION TABLE STANDARDS

700003 USAF
MILITARY STANDARDS
USAF,MIL-STD,1970
THESE DOCUMENTS GIVE DETAILED DESCRIPTIONS OF WHAT THE
MILITARY EXPECTS TO FIND WITH RESPECT TO ITS PROGRAMS AND
COMPUTER SYSTEMS. SPECIAL EMPHASIS IS ON VARIOUS FORMS AT
ALL LEVELS, AUTHORIZATION AND INTER-RELATION BETWEEN
PROJECTS. SOME OF THE 'WHY' OF DOCUMENTATION IS ALSO
INCLUDED.

700101 GOMAS, K.A.
OVERLAY TREE PROCESSOR
IBM CONTRIBUTED PROGRAM LIBRARY
THIS PROGRAM WILL CREATE A GRAPHIC REPRESENTATION OF THE
STRUCTURE OF AN OVERLAY PROGRAM AND OPTIONALLY RECREATE A
DECK OF SEQUENTIALLY NUMBERED CARDS, WHICH, THE LINKAGE
EDITOR WILL RECREATE THE SAME PRECISE STRUCTURE.

700301 BARNES, L.
RUNOFF: A PROGRAM FOR THE PREPARATION OF DOCUMENTS
NTIS-AD707402
THIS IS A USERS MANUAL FOR THE LANGUAGE RUNOFF. IT IS A
TIME-SHARING TEXT-EDITING DOCUMENT PREPARATION PROGRAM.

700401 MILLS, H.D.

700401 SYNTAX-DIRECTED DOCUMENTATION FOR PL360
COMM. ACM 13(APRIL 1970),216-222
PL360 IS USED AS BASIS FOR ILLUSTRATING IDEA CALLED
'SYNTAX-DIRECTED' DOCUMENTATION. IT USES SYNTACTIC TYPES
AND IDENTIFIERS TO TRIGGER AUTOMATIC FORMATION OF
QUESTIONS TO PROGRAMMER, WHOSE ANSWERS WILL BECOME PART
OF THE DOCUMENTATION. IT ALSO PROVIDES AUTOMATIC STORAGE
AND RETRIEVAL FACILITIES SO OTHER PROGRAMMERS CAN ACCESS
THE RESULTING DOCUMENTATION.

700601 MUTHUKRISHNAN, C.R.
ON THE CONVERSION OF DECISION TABLES TO COMPUTER PROGRAMS
COMM. ACM 13,6(JUNE 1970),347-351.
AUTHORS DISCUSS EXECUTION TIME DIAGNOSTICS AS A MEANS OF
POINTING OUT AMBIGUITIES IN DECISION TABLES. TWO
ALGORITHMS ARE PRESENTED FOR PROGRAMMING DECISION TABLES.
THE ALGORITHMS HAVE THE MERITS OF SIMPLICITY OF
IMPLEMENTATION AND DETECTION OF AMBIGUITIES AT EXECUTION
TIME. FEATURES OF A TRANSLATOR PREPARED BY THE AUTHORS ARE
ALSO DISCUSSED.

700602 IBM CORPORATION
IBM SYSTEM/360 OPERATING SYSTEM FULL AMERICAN NATIONAL
STANDARD COBOL PROGRAMMERS GUIDE NUMBERS 360S-CB-545
IBM CORPORATION
THIS MANUAL EXPLAINS HOW TO USE OS/360 TO COMPILE LINK-EDIT
AND EXECUTE PROGRAMS WRITTEN IN VERSION 2 OF IBM'S FULL
COBOL COMPILER.

700603 IBM CORPORATION
IBM SYSTEM/360 OPERATING SYSTEM FORTRAN IV (G AND H)
PROGRAMMERS GUIDE PROGRAM NUMBERS 360S-FO-520 360S-FO-520
IBM CORPORATION
THIS MANUAL EXPLAINS HOW TO USE OS/360 TO COMPILE LINK-EDIT
AND EXECUTE PROGRAMS WRITTEN IN IBM FORTRAN IV.

700604 FOGG, L.W.
CLOT - CARD LIBRARY ON TAPE
INTERNATIONAL TELECONTROL CORPORATION, WILMINGTON, DE (1970)
THIS BROCHURE DESCRIBES THE INPUT TO, CAPABILITIES OF, AND
OUTPUTS FROM THE TAPE-ORIENTED OFF-LINE PROGRAM EDITOR
'CLOT'. DEVICE AND STORAGE REQUIREMENTS, COST, AND SAMPLE
OUTPUT ARE ALSO GIVEN.

700701 BLUMBERG, S.E.
AN INTERIM PROGRESS REPORT OF COMPUTER OUTPUT MICROFILM
ACTIVITIES AND EXPERIENCES
NTIS-AD708600
THE PROCESS, COMPUTER-OUTPUT-MICROFILM(COM), IS DISCUSSED
WITH RESPECT TO ITS APPLICATION AT THE DEFENSE
DOCUMENTATION CENTER. ADVANTAGES AND COST-ANALYSIS ARE
EMPHASIZED.

700801 WICKER, R. NEPERLD, R. TEPLITZ, A.
MICROFICHE STORAGE AND RETRIEVAL SYSTEMS STUDY:
NTIS-AD710300
THE OBJECTIVE OF THE STUDY WAS TO DETERMINE USER REQUIREMENT
AND DEVELOP DESIGN OBJECTIVES AND SPECIFICATIONS FOR A

700801 LOW-COST MICROFICHE STORAGE AND RETRIEVAL SYSTEM. DESIGN OBJECTIVES AND SPECIFICATIONS FOR TWO OPTIMUM STORAGE AND RETRIEVAL DEVICES ARE PRESENTED.

700901 VARTABEDIAN, A.G.
THE DESIGN OF VISUAL DISPLAYS
BELL LABORATORIES REPORT 48, 8(SEPT 1970), 226.
VISUAL DISPLAYS APPEAL TO THE ECONOMY OF CRT DISPLAYS FOR MAN-MACHINE INTERPLAY. COMMUNICATION CAN BE INCREASED THROUGH CAREFULLY CHOOSING THE CARACTERISTICS OF THE DISPLAY SYMBOLS.

700902 DIAL, R.B.
DECISION TABLE TRANSLATION
COMM. ACM 13,9(SEPT 1970),571-573.
THE AUTHOR PRESENTS AN ALGORITHM FOR CONVERSION OF A LIMITED ENTRY DECISION TABLE INTO A MACHINE PROCESSABLE TEST-AND-BRANCH MATRIX. HE POINTS OUT THE RELATION OF THIS ALGORITHM TO OTHERS IN THE LITERATURE OF SIMILAR DESIGN

700903 STANFORD UNIVERSITY COMPUTATION CENTER
WYLBUR
STANFORD UNIVERSITY COMPUTATION CENTER, STANFORD, CA (1970)
THIS PACKAGE CONSISTS OF AN MVT MILTEN/WYLBUR OPERATOR'S GUIDE (MISNOMER--REALLY A USER REFERENCE MANUAL). WYLBUR IS AN ON-LINE EDITOR AND REMOTE JOB ENTRY AND RETRIEVAL SYSTEM USING SELECTRIC TYPEWRITER TERMINALS IN EITHER AN MFT OR MVT ENVIRONMENT. WYLBUR'S PROGRAM EDITING CAPABILITIES ARE TREATED AS A SUBSET OF ITS FREE-FORM TEXT EDITING CAPABILITIES. THE COVER LETTER WAS PRODUCED BY WYLBUR

700904 FELDMANN, C.G.
MAINTENANCE AND ENHANCEMENT OF THE AED SYSTEM
DDC-AD878211
(SEE AD875395 FOR DESCRIPTION)

700905 AMERICAN NATIONAL STANDARDS INSTITUTE, INC.
AMERICAN NATIONAL STANDARD FLOWCHART SYMBOLS AND THEIR USAGE IN INFORMATION PROCESSING
AMERICAN NATIONAL STANDARDS INSTITUTE INC, X395-1970, NY,NY, 1970
THIS DOCUMENT GIVES A DETAILED DESCRIPTION OF FLOWCHART, SYMBOL AND DECISION TABLE STANDARDS ACCEPTED BY ANSI.

701101 GOETZ, M.A.
AUTOFLOW ENHANCEMENTS OF DOCUMENTATION AND MAINTENANCE OF SCIENTIFIC APPLICATIONS
AUTOMATED METHODS OF COMPUTER PROGRAM DOCUMENTATION, (NOV 1970)
THE AUTHOR PROPOSES AND DISCUSSES THE FOLLOWING THREE ELEMENTS WHICH HE FEELS ARE ESSENTIAL TO PROVIDING A INTEGRATED DOCUMENTATION SYSTEM: 1)LOGICAL ANALYSIS OR GRAPHIC DISSECTION OF A PROGRAM(AUTOFLOW),2)HISTORY AND CONTROL OF PROGRAM(LIBRARIAN),3)UNDERSTANDING THE PROGRAM (TEXT EDIT PROGRAM).

701102 MOUTON, M.L.
DOCUMENTATION: MOTIVATION AND TRAINING OR AUTOMATION

701102 AUTOMATED METHODS OF COMPUTER PROGRAM DOCUMENTATION, (NOV 1970)
THE AUTHOR DESCRIBES WHAT MANUALS NEED TO BE DEVELOPED WHILE DESIGNING A SYSTEM, BY WHOM AND FOR WHOM THEY ARE DESIGNED, AND EMPHASIZES THE NEED FOR DEVELOPING SYSTEM DOCUMENTS BEFORE SYSTEM IS FINISHED.

701103 WOLF, A.W.
MAKING AUTOMATED COMPUTER PROGRAM DOCUMENTATION A FEATURE OF TOTAL SYSTEM DESIGN
AUTOMATED METHODS OF COMPUTER PROGRAM DOCUMENTATION, (NOV 1970)
THE AUTHOR NOTES THAT DOCUMENTS ARE OFTEN THE AFTERMATH OF SYSTEMS DESIGNS. HE DISCUSSES THE DESIGN OF A NEW SOFTWARE SYSTEM BY AFSCF FOR THE AIR FORCE, AND THE PROBLEMS OF DOCUMENTATION. OF SPECIAL INTEREST IS THE DISCUSSION OF THE DATA BASE DESCRIPTION CALLED 'COMPOOL' FOR COMMON POOL OF INFORMATION.

701104 NEELY, M.D. TYSON, J.W.
AUTOMATIC PROGRAM ANNOTATION (AUTONOTE)
AUTOMATED METHODS OF COMPUTER PROGRAM DOCUMENTATION, (NOV 1970)
THIS PAPER DESCRIBES A HYPOTHETICAL AUTOMATIC DOCUMENTATION SYSTEM BASED ON SUPPLEMENTING RATHER THAN REPLACING THE PROGRAMMER COMMENTS IN THE SOURCE DECK. IT IS VERY DETAILED.

701105 PARDEE, S.
BELLFLOW AUTOMATIC FLOWCHARTING SYSTEM
AUTOMATED METHODS OF COMPUTER PROGRAM DOCUMENTATION, (NOV 1970)
THE BELLFLOW SYSTEM IS DISCUSSED WITH RESPECT TO THE THREE MODES OF OPERATION. THEY ARE CALLED THE SOURCE MODE, COMMENT MODE, AND MIXED MODE. THE BELLFLOW SYSTEM ALLOWS FOR USEFUL AND MEANINGFUL COMMENTS TO DERIVE A AUTOMATIC FLOWCHART. BELLFLOW IS A TABLE-DRIVEN SYSTEM.

701106 HANNEY, R.
AN AUTOMATED SYSTEM FOR GENERATING PROGRAM DOCUMENTATION
AUTOMATED METHODS OF COMPUTER PROGRAM DOCUMENTATION, (NOV 1970)
THE AUTHOR DEVELOPED A DOCUMENTATION PROGRAM WITH EMPHASIS PLACED ON TEXT CONTENT RATHER THAN FLOWCHARTING. THE PROGRAM GENERATES THE ENTIRE DOCUMENT AND IT IS KEYWORD ORIENTED.

701107 GROSCH, H.R.
VIEWS ON COMPUTER PROGRAM DOCUMENTATION AND AUTOMATION
AUTOMATED METHODS OF COMPUTER PROGRAM DOCUMENTATION, (NOV 1970)
DR. GROSCH EXPRESSES SOME BROAD OPINIONS AND PHILOSOPHY ABOUT THE PROBLEMS OF DOCUMENTATION AND DESCRIPTION. HE CAUTIONS ABOUT THE SEMANTICS ON THE SUBJECT OF DOCUMENTATION. SOME ESTIMATED COSTS OF COMPUTING AND DATA PROCESSING IN THE FEDERAL ESTABLISHMENT ARE ALSO GIVEN.

701108 MCCLURE, C.W.

701108 AUTOMATED ENGINEERING DESIGN AN APPROACH TO AUTOMATED DOCUMENTATION
AUTOMATED METHODS OF COMPUTER PROGRAM DOCUMENTATION, (NOV 1970)
THE AUTOMATED ENGINEERING DESIGN SYSTEM IS A SYSTEM OF COMPUTER PROGRAMS DESIGNED FOR USE IN BUILDING SOFTWARE SYSTEMS. DOCUMENTATION (ANYTHING THAT HELPS A PERSON UNDERSTAND A PROGRAM) IS A NECESSARY PART OF SOFTWARE DEVELOPMENT BECAUSE THE SOFTWARE WILL INVARIABLY UNDERGO CHANGES OVER THE YEARS.

701109 LANZANO, B.C.
PROGRAM AUTOMATED DOCUMENTATION METHODS
AUTOMATED METHODS OF COMPUTER PROGRAM DOCUMENTATION, (NOV 1970)
THIS PAPER PRESENTS A DESCRIPTION OF A PROGRAM WHICH ASSISTS IN AUTOMATING THE DOCUMENTATION OF SUBROUTINES(ADS), AN EXPOSE OF TWO FLOWCHARTING PROGRAMS(AUTOFLOW,FLOWGEN). SOME NOTES ON USEFUL PROGRAM INTERNAL CROSS-REFERENCE INFORMATION, A TEXT-EDITING PROGRAM FOR A TIME-SHARED ENVIRONMENT, A SYSTEM TO AID PROGRAM DOCUMENTATION UTILIZING A GRAPHICS DISPLAY CONSOLE.

701110 THOMAS, V.L.
AUTOMATED DOCUMENTATION OF AN ASSEMBLY PROGRAM
AUTOMATED METHODS OF COMPUTER PROGRAM DOCUMENTATION, (NOV 1970)
THIS PAPER IS AN AUTOMATIC DOCUMENTATION PROPOSAL WHICH EXTRACTS COMMENTS FROM CODE AND OUTPUTS IT IN AN ORDERLY LIST SEPARATE FROM THE LISTING.

701111 KALAF, M.C.
COSMIC: PROGRAM DOCUMENTATION EXPERIENCE
AUTOMATED METHODS OF COMPUTER PROGRAM DOCUMENTATION, (NOV 1970)
THE PAPER DESCRIBES THE MINIMUM DOCUMENTATION EFFORT REQUIRED IN ORDER TO DISEMENATE PROGRAMS TO USERS IN A USABLE FORM.

701112 RICH, R.P.
AUTOMATIC EDITING OF MANUALS
BRIEFLY DISCUSSES EDITING SYSTEM (INFO 360) AND PROVIDES AN OVERVIEW OF IT. ALSO PROPOSES A METHOD OF DOCUMENTATION CALLED 'MONODOCUMENTATION' WHICH IS A 'SUPER-LISTING' OF A PROGRAM. THIS METHOD WOULD HAVE ALL RELEVANT DOCUMENTA-TION NEEDED FOR DESCRIBING THE PROGRAM MAINTAINED WITH IT. WHEN THE PROGRAM IS UPDATED OR CHANGED THE APPROPRIATE DOCUMENTATION IS UPDATED OR CHANGED ALSO.

701113 FELSMAN, W.O.
COST ADVANTAGES OF AN INTEGRATED DOCUMENTATION APPROACH
AUTOMATED METHODS OF COMPUTER PROGRAM DOCUMENTATION (NOV 1970)
THE USE OF METAPROGRAMS IS ADVOCATED TO REDUCE PROGRAMMING TIME AND ASSOCIATED DOCUMENTATION PREPARATION. GENERIC PROGRAMS SHOULD BE WRITTEN WHICH CAN BE TAILORED BY MEANS OF A METAPROGRAM WHICH PROCESSES THE PARAMETERS TO PERFORM A SPECIFIC FUNCTION. DOCUMENTATION COST CAN BE FURTHER RE-

701113 DUCED THROUGH THE USE OF SMALL PREDOCUMENTED SUBPROGRAMS.

701114 LUEBKE, W.R.
THE INTEGRATION OF SYSTEM SPECIFICATIONS AND PROGRAM CODING. A REPORT ON EXPERIENCE OF THE MEDLARS II PROJECT.
AUTOMATED METHODS OF COMPUTER PROGRAM DOCUMENTATION (NOV 1970)
DESCRIBES CSC EXPERIENCE IN MAINTAINING UP-TO-DATE DOCUMENTA TION FOR ONE MODULE OF A VERY LARGE SCALE PROJECT,MEDLARS. SEVERAL INNOVATIVE TECHNIQUES HAVE BEEN EXPLORED IN THE CONTENT OF MEDLARS' DATA MANAGEMENT ENVIRONMENT USING PL/I AS AN AUTOMATIC DOCUMENTER. PL/I'S DATA DESCRIPTION PROVIDES AUTOMATIC DOCUMENTATION USING 'MASTER DESCRIPTION ' OF DATA ELEMENTS,WITH MEANINGFUL NAMES.

701201 MENKUS, B.
DEFINING ADEQUATE SYSTEMS DOCUMENTATION
J SYSTEMS MGMT 21,12(DEC 1971),16-21.
AUTHOR DEFINES SYSTEMS DOCUMENTATION AND ENUMERATES FIVE FUNCTIONS PERFORMED BY ADEQUATE SYSTEMS DOCUMENTATION. HE ALSO LISTS SEVERAL IMPORTANT GENERAL CONSIDERATIONS ABOUT DOCUMENTATION. CATEGORIZES TWO AREAS, SYSTEMS DEVELOPMENT DOCUMENTATION, AND SYSTEMS OPERATION DOCUMENTATION UNDER THE GENERAL HEADING OF SYSTEMS DOCUMENTATION. ALSO CONSIDERS STANDARDS.

701202 BERKELEY, E.C.
RESEARCH IN COMPUTER-ASSISTED DOCUMENTATION OF COMPUTER PROGRAMS
NTIS-AD719451
DISCUSSES THE PROBLEM OF DOCUMENTATION OF COMPUTER PROGRAMS. PRESENTS A FURTHER MODEL OF A 'SIMULATOR-ANALYZER' COMPUTER PROGRAM. A COMPLETE COMPUTER PROGRAM FOR AN SA IS ALSO PRESENTED. HE DISCUSSES COMMENTS AND MNEMONIC SYMBOLS IN COMPUTER PROGRAMS.

701203 IBM CORPORATION
IBM SYSTEM/360 OPERATING SYSTEM ASSEMBLER F PROGRAMMERS GUIDE PROGRAM NUMBER 360L-AS-037
IBM CORPORATION
THIS MANUAL DISCUSSES PROGRAM ASSEMBLING, LINKAGE EDITING, EXECUTING, INTERPRETING LISTINGS, ASSEMBLER PROGRAMMING CONSIDERATIONS, DIAGNOSTIC MESSAGES, AND OBJECT OUTPUT CARDS.

710101 IBM CORPORATION
IBM SYSTEM/360 OPERATING SYSTEM PL/I(F) PROGRAMMERS GUIDE PROGRAM NUMBER 360S-NL-511
IBM CORPORATION
THIS MANUAL EXPLAINS HOW TO USE OS/360 TO COMPILE LINK-EDIT AND EXECUTE PROGRAMS WRITTEN IN IBM'S VERSION OF PLII.

710102 SPERRY RAND CORPORATION
UNIVAC 1100 SERIES OPERATING SYSTEM
UNIVAC 1100 SERIES OPERATING SYSTEM NUMBER 4144 REV.2
THIS MANUAL DISCUSSES THE BASE PORTION OF THE OPERATING SYSTEM (EXEC 8) AND THE ASSOCIATED SOFTWARE NEEDED TO CONSTRUCT, EXECUTE, AND MAINTAIN USER PROGRAMS.

710103 SPERRY RAND CORPORATION
ED PROCESSOR - UNIVAC 1100 SERIES OPERATING SYSTEM PROGRAMMER REFERENCE
SPERRY RAND CORPORATION, (1971)
THE 'ED' PROCESSOR IS AN OFF-LINE TEXT EDITOR OPERATING UNDER THE EXEC-8 EXECUTIVE SYSTEM ON UNIVAC'S 1100 SERIES COMPUTERS. FROM THE MANUAL'S EXPLANATION, 'ED' APPEARS CAPABLE OF OPERATING EITHER IN OFF-LINE OR ON-LINE MODE.

710104 CONTROL DATA CORPORATION
EDIT - KRONOS TEXT EDITOR (EDIT) REFERENCE MANUAL
CONTROL DATA CORPORATION, MINNEAPOLIS, MINN (1971)
THIS REFERENCE MANUAL DESCRIBES THE ON-LINE TEXT EDITOR, 'EDIT', PART OF THE KRONOS TIME-SHARING SYSTEM. 'EDIT' ALLOWS MERGING OF MULTIPLE FILES AND ELIMINATION OF ANY EXCESS BLANKS. IT HANDLES FILES OF UP TO 250,000 LINES, EACH OF UP TO 150 CHARACTERS.

710105 CONTROL DATA CORPORATION
KRONOS TEXT EDITOR (EDIT)
CONTROL DATA CORPORATION
TEXT EDITOR ALLOWS THE USER TO EDIT A DATA FILE. THE DATA FILE BEING EDITED IS KNOWN AS THE SEARCH FILE. DURING EDITING, THE SEARCH-POINTER IDENTIFIES THE LINE OF THE SEARCH FILE THAT IS CURRENTLY ACCESSIBLE. THE SEARCH POINTER CAN BE MOVED FORWARD AND BACKWARD DURING EDITING TO SPECIFY A NEW LINE.

710201 SCHWAYDER, K.
CONVERSION OF LIMITED-ENTRY DECISION TABLES TO COMPUTER PROGRAMS A PROPOSED MODIFICATION TO POLLACKS ALGORITHM
COMM. ACM 14(FEB 1971),69-73
MODIFICATIONS ARE PROPOSED TO POLLACKS ALGORITHM, WHICH MINIMIZE SUBSEQUENT EXECUTION TIME WHEN COMPILED INTO A COMPUTER PROGRAM.

710301 CHAPIN, N.
PERSPECTIVE ON FLOWCHART PACKAGES
COMPUTERS AND AUTOMATION 20, 3(MARCH 1971),16-19.
A COMPARATIVE STUDY OF VARIOUS AUTOMATIC FLOWCHARTING PACKAGES WITH RESPECT TO FACTORS SUCH AS SPEED, DESIGN, AND COST.

710302 GEORGE, R.L.
AUTOMATIC SELECTIVE DOCUMENTATION SERVICES
NTIS-AD722425
DDC HAS BEEN DEVELOPING AND TESTING SYSTEMS BASED ON THE APPROACH OF DETERMINING USERS SPECIFIC DOCUMENTATION REQUIREMENTS AND AUTOMATICALLY DISSEMINATING THE NEEDED DOCUMENTATION TO THE USERS' LOCAL LIBRARIES AS SOON AS IT BECOMES AVAILABLE. THIS CONCEPT IS APPLIED TO THE SELECTIVE DISSEMINATION OF BOTH REPORT ANNOUNCEMENTS AND FULL-TEXT REPORTS.

710303 WILLIAMS, R.
A SURVEY OF DATA STRUCTURES FOR COMPUTER GRAPHICS SYSTEMS
NTIS-AD725284
REASONS FOR USING DATA STRUCTURES ARE GIVEN. THE SEQUENTIAL

710303 RANDOM, AND LIST ORGANIZATIONS ARE DISCUSSED, AND IT IS SHOWN HOW THEY MAY BE USED TO BUILD COMPLEX DATA STRUCTURES. REPRESENTATIVE SAMPLES OF LANGUAGES SPECIFICALLY DESIGNED FOR CREATING AND MANIPULATING DATA STRUCTURES ARE DESCRIBED.

710601 GUNDERMAN, R.E.
HARD LOOK AT SOFTWARE DOCUMENTATION
J. SYSTEMS MGMT 21,6(JUNE 1971),35-36.
AUTHOR PROPOSES A SET OF CRITERIA FOR EVALUATING PROGRAMMING DOCUMENTATION IN ORDER TO ESTABLISH A BASIS FOR OVERALL IMPROVEMENT OF DOCUMENTATION.

710602 BRACKETT, J.W.
MAINTENANCE AND ENHANCEMENT OF THE AED SYSTEM
DDC-AD887141
(SEE AD875395 FOR DESCRIPTION)

710603 INTERACTIVE SCIENCES CORPORATION
TECO - TEXT EDITOR AND CORRECTOR - REFERENCE MANUAL
INTERACTIVE SCIENCES CORPORATION, BRAINTREE, MA (1971)
THIS REFERENCE MANUAL DESCRIBES BOTH BASIC AND ADVANCED TECHNIQUES IN USING TECO ON A DEC PDP-10 COMPUTER. THIS ON-LINE PROGRAM EDITOR ALLOWS EDITING TO BE DONE NOT ONLY BY SINGLE COMMANDS, BUT ALSO BY SMALL 'PROGRAMS' FOR MORE COMPLEX EDITING.

710901 ELLIS, T.O.
ARPA NETWORK SERIES I.INTRODUCTION TO THE ARPA NETWORK AT RAND AND TO THE RAND VIDEO GRAPHICS SYSTEM
NTIS-AD733049
AN OVERVIEW OF THE ADVANCED RESEARCH PROJECTS AGENCY'S EXPERIMENTAL COMPUTER NETWORK, AND A TECHNICAL DESCRIPTION OF THE RAND VIDEO GRAPHIC SYSTEM THAT LINKS RAND COMPUTING RESOURCES INTO THE NETWORK.

710902 WELLS, E.E.
A LIBRARY OF DATA COLLECTION AND MANIPULATION SUBROUTINES .
NTIS-AD734330
PROGRAMS TO ALLOW THE MINICOMPUTER TO PERFORM SOPHISTICATED DATA COLLECTION,CONTROL,AND MANIPULATION FUNCTIONS IN A LABORATORY ENVIRONMENT ARE PRESENTED. THE EMPHASIS OF THE LIBRARY IS UPON FLEXIBILITY OF USE AND EASE OF CHAINING THE SUBROUTINES TO MAKE USEFUL PROGRAMS. IN ADDITION TO FLOW CHARTS WHICH INDICATE THE ALGORITHM,ACTUAL PROGRAM LISTINGS ARE PROVIDED.

711001 DEFENSE DOCUMENTATION CENTER
MICROFICHE, MICROFILM, AND RELATED EQUIPMENT
NTIS-AD732800
THIS IS A BIBLIOGRAPHY OF MICROFICHE, MICROFILM AND RELATED EQUIPMENT. ENTRIES HAVE BEEN SELECTED FROM REFERENCES PROCESSED INTO THE AD DATA BANK FROM JAN. 1968 TO JULY 1971.

711002 KREUTZER, P.J.
DATA COMPRESSION IN LARGE BUSINESS-ORIENTED FILES
NTIS-AD734394

711002 DATA COMPRESSION IS OF INTEREST IN DATA PROCESSING BECAUSE IT OFFERS COST SAVINGS AND THE POTENTIAL FOR INCREASED CAPACITY IN MASS STORAGE DEVICES,CHANNELS AND COMMUNICATIONS LINES,ALSO THESE TECHNIQUES ARE REVIEWED AS THEY APPLY TO BUSINESS DATA FILES AND AN IMPLEMENTATION OF DATA COMPRESSION AN A BUSINESS TYPE DRUM FILE UNDER SEVERE CONSTRAINTS OF CORE,EXECUTION TIME,AND COMPRESSION REQUIREMT.

711101 TATMAN, J.C.
ACHIEVING PROPER PROGRAM DOCUMENTATION
J. SYSTEMS MGMT 21,11(NOV 1971), 40-41
A PROCEDURE IS PRESENTED FOR HELPING TO PRODUCE PROPER PROGRAM DOCUMENTATION WITHIN AN ORGANIZATION. EMPHASIS IS PLACED ON ADOPTION OF STANDARDS. PROCEDURE INCLUDES MANAGEMENT APPROVAL, DEVELOPMENT OF STANDARDS, ASSIGNMENT OF RESPONSIBILITIES, AND IMPLEMENTATION.

720101 SIGMA SOFTWARE COMPANY
CFMS - CARD FILE MAINTENANCE SYSTEM
SIGMA SOFTWARE COMPANY, GUTHRIE, OK, 1972
THIS BROCHURE DESCRIBES THE THREE COMPONENTS OF THIS OFF-LINE PROGRAM EDITOR - AN UPDATE PROGRAM, AN INDEX REPORT GENERATOR, AND A RETRIEVAL PROGRAM. CFMS IS A TAPE-ORIENTED EDITOR OF VERY LOW PRICE AND RATHER LIMITED CAPABILITIES.

720102 USAF
DOCUMENTATION STANDARDS
DEPARTMENT OF THE AIR FORCE, COMMUNICATIONS COMPUTER PROGRAMMING CENTER, TINKER AFB, OKLA, 10 JAN 1972
THIS DOCUMENT BRIEFLY DESCRIBES WHY DOCUMENTATION IS NEEDED, DISCUSSES SYSTEMS FLOWCHARTS AND VARIOUS ANS REQUIREMENTS ON FLOWCHARTS. SAMPLE FLOWCHARTS AND A PROGRAM ABSTRACT ARE INCLUDED. THE MAJORITY OF THE REST OF THIS DOCUMENT DEALS SPECIFICALLY WITH AIR FORCE IDENTIFICATION CODES AND IS OF LITTLE INTEREST. A PAGE OF DEFINITIONS OF COMPUTER TERMS IS INCLUDED.

720103 GODDARD SPACE FLIGHT CENTER
STANDARDS GUIDE FOR SPACE AND EARTH SCIENCES COMPUTER SOFTWARE
GODDARD SPACE FLIGHT CENTER, GREENBELT,MARYLAND,JANUARY,1972
THIS DOCUMENT IS A SET OF GUIDELINES FOR DOCUMENTATION RECOMMENDED (BUT NOT REQUIRED) FOR USE AT GSFC. EMPHASIS IS ON LANGUAGE STANDARDS AND APPLICATION OF GOOD MANAGEMENT TECHNIQUES. IT CONTAINS PROGRAMMING STANDARDS, DOCUMENTATION STANDARDS, TESTING AND ACCEPTANCE TECHNIQUES , CORRECTION AND UPDATE STANDARDS.

720201 CULLINANE CORPORATION
PLUS D/A, A DIRECT ACCESS SOFTWARE SECURITY AND CONTROL SYSTEM
CULLINANE CORPORATION, BOSTON, MA, 1972
THIS BROCHURE DESCRIBES THE INPUT TO, OUTPUT FROM, AND FEATURES OF THE OFF-LINE PROGRAM EDITOR, PROGRAM LIBRARY UPDATE SYSTEM DIRECT ACCESS (PLUS D/A). ITS PRIMARY SECURITY MEASURE IS A CHANGEABLE, SCRAMBLED CHARACTER SET. THIS PROGRAM EDITOR IS ALSO AVAILABLE IN A TAPE-ORIENTED VER-

720201 SION, KNOWN SIMPLY AS 'PLUS'.

720501 BUSINESS AUTOMATION
TALKING DOWN A PROGRAM
BUSINESS AUTOMATION 19,5(MAY 1972), 26-27.
THIS ARTICLE DESCRIBES THE EXPERIENCE OF A SW BELL GROUP WHICH DICTATES COBOL CODE INSTEAD OF WRITING IT LONGHAND. THE TECHNIQUE WAS FOUND TO INCREASE PROGRAMMER EFFICIENCY AND REDUCE ERRORS.

720502 PANSOPHIC SYSTEMS, INC.
PANVALET - THE DIRECT ACCESS LIBRARY SYSTEM
PANSOPHIC SYSTEMS, INC., OAK BROOK, IL (1972)
THIS PACKAGE CONTAINS AN OS USER REFERENCE MANUAL, AN OS SYSTEM MANAGEMENT MANUAL, AND AN EXTENSIVE SAMPLE OF THE OUTPUT FROM PANVALET. THIS OFF-LINE PROGRAM EDITOR SUPPORTS UP TO 3 LEVELS OF SECURITY CONTROL CODES AND ALLOWS TEMPORARY UPDATING AND THE CAPABILITY OF CROSS REFERENCING LIBRARY DATA SETS

720503 XEROX CORPORATION
XEROX TEXT LANGUAGE AND OPERATIONS REFERENCE MANUAL
XEROX CORPORATION
TEXT CONSISTS OF A CENTRAL XEROX COMPUTER (MODEL SIGMA 6,7, OR 9) OPERATING UNDER UNIVERSAL TIME-SHARING SYSTEM (UTS), THE TEXT PROGRAM, AND ONE OR MORE REMOTE TYPEWRITER TERMINALS CONNECTED TO THE COMPUTER VIA TELEPHONE LINES.

720601 APPLIED DATA RESEARCH, INC.
LIBRARIAN
APPLIED DATA RESEARCH, INC., PRINCETON, NJ, 1972
THIS PACKAGE CONTAINS A CONCEPTS AND FACILITIES FOR EDP MANAGERS MANUAL, A USER REFERENCE MANUAL, AN OS SYSTEM REFERENCE MANUAL, AND A DESCRIPTION OF THE SPACESAVER DISK MANAGEMENT FEATURE. LIBRARIAN IS AN OFF-LINE PROGRAM EDITOR WITH AN OPTIONAL COBOL SYNTAX CHECKER AND SOME ADAPTABILITY TO A CONVERSATIONAL TERMINAL OPERATIONAL ENVIRONMENT.

720602 MANAGEMENT SYSTEMS CORPORATION
PROGRAM/MANAGE, A PROGRAM TO MAINTAIN SOURCE PROGRAM
MANAGEMENT SYSTEMS CORPORATION, DALLAS, TX
THIS BROCHURE DESCRIBES THE INPUT TO, CAPABILITIES OF, AND OUTPUTS FROM THE TAPE-ORIENTED OFF-LINE PROGRAM EDITOR 'PROGRAM/MANAGE'. DEVICE AND STORAGE REQUIREMENTS AND COST ARE ALSO GIVEN.

720603 AUERBACH ASSOCIATES, INC.
DATA MANIPULATION
AUERBACH COMPUTER TECHNOLOGY REPORTS, SOFTWARE REPORTS, 51, (1972), 6007.85-.87, .91-.93, .107-.108, 6008.129-.131.
THIS SECTION DESCRIBES THE INPUT TO, PROCESSING OF, AND OUTPUTS FROM THE OFF-LINE PROGRAM EDITORS 'LIBRARIAN', 'PLUS D/A', 'SPLIS-II', AND 'SIMPLE' (WHICH SEE). IT GIVES DEVICE AND STORAGE REQUIREMENTS, SOURCE LANGUAGE, COST, AND SPECIAL FEATURES OF EACH OF THE ABOVE PROGRAM EDITORS.

720604 VAN-DAM, A. RICE, D.E.
ON-LINE TEXT EDITING: A SURVEY

720604 ACM COMPUTING SURVEYS 3, 3(SEP 1971),93-114
THIS ARTICLE DISCUSSES THE ADVANTAGES OF ON-LINE EDITING OF BOTH COMPUTER PROGRAMS AND FREE-FORM TEXT, THE DISTINGUISHING CHARACTERISTICS OF TWO TYPES OF ON-LINE EDITORS, AND THE CAPABILITIES OF SEVERAL WORKING SYSTEMS OF BOTH TYPES. A RATHER COMPREHENSIVE BIBLIOGRAPHY FOLLOWS THE ARTICLE.

720605 COMPRESS, INC.
COMCHART
COMPRESS
AUTOMATICALLY DOCUMENTS COBOL AND ASSEMBLEY SOURCE PROGRAMS. IT CAN ALSO BE USED WITH A SPECIAL DESIGN LANGUAGE TO HELP IN PROGRAM DEVELOPMENT. COMCHART PRODUCES A DETAILED FLOWCHART AND A NUMBER OF CROSS-REFERENCES.

720606 MANTECH CORPORATION
SUPEREF
MANTECH CORPORATION
SUPEREF AUTOMATICALLY PRODUCES A CROSS-REFFRENCE LISTING OF FORTRAN SOURCE PROGRAMS. NO FLOWCHARTS ARE GIVEN.

720607 DATA FOR MANAGEMENT DECISION
FORTRAN VARIABLE NAME DOCUMENTER
DATA FOR MANAGEMENT DECISION
PRODUCES A SOURCE LISTING AND ALPHABETICAL LIST OF REFERENCED VARIABLE NAMES.

720608 DNA SYSTEMS, INC.
FORFLO
DNA SYSTEMS, INC.
FORFLO AUTOMATICALLY DOCUMENTS FORTRAN SOURCE PROGRAMS. IT PRODUCES DETAILED FLOWCHART AND A RESEQUENCED SOURCE LISTING.

720609 CALCOMP
FLOWGEN/F
CALCOMP
FLOWGEN/F AUTOMATICALLY PRODUCES A DETAILED FLOWCHART FROM A FORTRAN SOURCE PROGRAM. IT PRODUCES THE FLOWCHART ON A CALCOMP PLOTTER.

720610 APPLICATIONS PROGRAMMING CO.
DYNACHART
APPLICATIONS PROGRAMMING CO.
DYNACHART AUTOMATICALLY DOCUMENTS COBOL SOURCE PROGRAMS. IT PRODUCES A DETAILED FLOWCHART AND A CROSS-REFERENCE LISTING.

720611 NATIONAL COMPUTER ANALYSIS, INC.
QUICK-DRAW
NATIONAL COMPUTER ANALYSTS
QUICK-DRAW AUTOMATICALLY DOCUMENTS COBOL, FORTRAN, AND ASSEMBLY SOURCE PROGRAMS, IT PRODUCES A DETAILED FLOWCHART, CROSS-REFERENCES, AND SOME SPECIAL LISTINGS.

720612 APPLIED DATA RESEARCH
AUTOFLOW

720612 APPLIED DATA RESEARCH, INC.
AUTOMATICALLY DOCUMENTS COBOL,PL/I,ASSEMBLY,AND FORTRAN SOURCE PROGRAMS. AUTOFLOW PRODUCES FLOWCHARTS OF DIFFERENT LEVELS OF DETAIL,CROSS-REFERENCES,AND MANY SPECIAL LISTINGS.

720613 BELOIT COMPUTER CENTER
AUTOCHART
BELOIT COMPUTER CENTER
DOCUMENTS COBOL SOURCE PROGRAMS. PRODUCES A DETAILED FLOWCHART AND A CROSS-REFERENCE LISTING OF ALL LABELS USED IN THE PROGRAM.

720614 AIRES CORP.
AUTODIAGRAMMER II
AIRES CORP.
AUTOMATICALLY DOCUMENTS COBOL SOURCE PROGRAMS. PRODUCES A DETAILED FLOWCHART, A DIAGNOSTIC LISTING, AN ALTERED STATEMENT CROSS-REFERENCE, A RECORD LAYOUT, AND A UNMATCHED LABEL TABLE.

720615 DATA INSTRUMENTS CO.
AUTODOC
DATA INSTRUMENTS CO.
AUTOMATIC DOCUMENTATION OF COBOL AND ASSEMBLY SOURCE PROGRAMS. PRODUCES A DETAILED FLOWCHART AND CROSS-REFERENCES. IT CAN ALSO OPTIONALLY PRODUCE A LOGIC CHART.

720616 SYSTONETICS CORP.
EZFLOW
SYSTONETICS CORP.
AUTOMATIC DOCUMENTATION OF FORTRAN SOURCE PROGRAMS. PRODUCES A DETAILED FLOWCHART AND CROSS-REFERENCES. EZFLOW ALSO CAN RESEQUENCE SOURCE CODE LABEL NUMBERS.

720617 TEXT SYSTEMS,INC.
FRESS USER'S GUIDE
TEXT SYSTEMS,INC.
FRESS FILE RETRIEVAL AND EDITING SYSTEM) IS A SOPHISTICATED AND COST EFFECTIVE TEXT MANIPULATION SYSTEM. IT IS A VASTLY ENHANCED, MULTICONSOLE PRODUCTION VERSION OF THE PROTOTYPE HES (HYPERTEXT EDITING SYSTEM). IT IS AN ONLINE SYSTEM USING IBM 2741 OR 2260 TERMINALS OR EQUIVALENT.

720701 ARPA NETWORK INFORMATION CENTER
NETWORK INFORMATION CENTER USER GUIDE
AUGMENTATION RESEARCH CENTER, STANFORD RESEARCH INSTITUTE, MENLO PARK, CA, JULY, 1972
CONTENTS:1.SAMPLE MESSAGE SENDING SESSION. 2.NIC TNLS USER GUIDE. 3.NIC JOURNAL SYSTEM USER GUIDE. 4.LOCATOR. 5.PHONE SERVICE. 6.FOLKLORE DESCRIBES HOW TO USE THE FACILITIES OF THE ARPA NETWORK INFORMATION CENTER.

720702 INFORMATION CONTROL SYSTEMS,INC.
ASTROCOMP
INFORMATION CONTROL SYSTEMS,INC.
THIS SYSTEM PROVIDES TEXT EDITING CAPABILITIES ON MAGNETIC TAPE. WIDE VARITY OF OPTIONS ARE ALSO AVAILABLE.

ASTROCOMP

ASTROCOMP 720702

AUTOCHART

AUTOCHART NEW METHOD CUTS MANUFACTURING. 640501

AUTOCHART 720613

AUTODIAGRAMMER

AUTODIAGRAMMER II 720614

AUTODOC

AUTODOC: COMPUTER-BASED ASSISTANCE FOR DOCUMENT PRODUCTION 670001

AUTODOC 720615

AUTOFLOW

AUTOFLOW ENHANCEMENTS OF DOCUMENTATION AND MAINTENANCE OF SCIENTIFIC APPLICATIONS 701101

AUTOFLOW 720612

AUTOMATED

AUTOMATED WEATHER NETWORK 000101

AUTOMATED PLOTTING OF FLOWCHARTS ON A SMALL COMPUTER 650101

MAKING AUTOMATED COMPUTER PROGRAM DOCUMENTATION A FEATURE OF TOTAL SYSTEM DESIGN 701103

AN AUTOMATED SYSTEM FOR GENERATING PROGRAM DOCUMENTATION 701106

AUTOMATED ENGINEERING DESIGN AN APPROACH TO AUTOMATED DOCUMENTATION 701108

PROGRAM AUTOMATED DOCUMENTATION METHODS 701109

AUTOMATED DOCUMENTATION OF AN ASSEMBLY PROGRAM 701110

AUTOMATIC

AUTOMATIC PREPARATION OF FLOWCHART LISTING. 580101

AUTOMATIC DATA PROCESSING SUPPORT FOR FLAG

AUTOMATIC CONTINUATION

PLOT 660402

AUTOMATIC PROGRAM ANNOTATION (AUTONOTE) 701104

BELLFLOW AUTOMATIC FLOWCHARTING SYSTEM 701105

AUTOMATIC EDITING OF MANUALS 701112

AUTOMATIC SELECTIVE DOCUMENTATION SERVICES 710302

AUTOMATION

AUTOMATION IN LINGUISTICS 680103

DOCUMENTATION: MOTIVATION AND TRAINING OR AUTOMATION 701102

VIEWS ON COMPUTER PROGRAM DOCUMENTATION AND AUTOMATION 701107

AUTONOTE

AUTOMATIC PROGRAM ANNOTATION (AUTONOTE) 701104

BELL

THE GROWTH OF COMPUTER GRAPHICS AT BELL LABORATORIES. 680601

BELLFLOW

BELLFLOW AUTOMATIC FLOWCHARTING SYSTEM 701105

BERRER

DECISION TABLES FOR BERRER MANAGEMENT SYSTEMS 660301

BIBLIOGRAPHY

DIAGNOSTIC PROGRAMMING FOR DIGITAL COMPUTERS A BIBLIOGRAPHY 670402

BIBLIOGRAPHY ON INFORMATION SCIENCES. VOLUME I 680201

BURDEN

PROGRAM DOCUMENTATION: THE VALUABLE BURDEN 680501

BUSINESS-ORIENTED

DATA COMPRESSION IN LARGE

A-40

CONVERSION
CONTINUATION

ON THE CONVERSION OF DECISION TABLES TO COMPUTER PROGRAMS	700601
CONVERSION OF LIMITED-ENTRY DECISION TABLES TO COMPUTE PROGRAMS A PROPOSED MODIFICATION TO POLLACKS ALGORITHM	710201

CORRECTOR

TECO - TEXT EDITOR AND CORRECTOR - REFERENCE MANUAL	710603

CODING

THE INTEGRATION OF SYSTEM SPECIFICATIONS AND PROGRAM CODING. A REPORT ON EXPERIENCE OF THE MEDLARS II PROJECT.	701114

COSMIC

COSMIC: PROGRAM DOCUMENTATION EXPERIENCE	701111

CROSS-REFERENCE

CONTROL SECTION CROSS-REFERENCE AND LINK EDIT EDITOR	681003

DECISION

USING DECISION STRUCTURE TABLES	640301
USE OF DECISION TABLES IN COMPUTER PROGRAMMING	650102
CONVERSION OF LIMITED ENTRY DECISION TABLES TO COMPUTER PROGRAMS	651101
DATA, DOCUMENTATION, AND DECISION TABLES	660101
PROGRAMMING DECISION TABLES IN FORTRAN, COBOL OR ALGOL	660102
DECISION TABLES FOR BERRER MANAGEMENT SYSTEMS	660301
CONVERSION OF LIMITED-ENTRY DECISION TABLES TO OPTIMAL COMPUTER PROGRAMS I: MINIMUM AVERAGE PROCESSING TIME	660801
CONVERSION OF DECISION TABLES TO COMPUTER PROGRAMS BY RULE MASK TECHNIQUES	661101
DECISION TABLES	670802

DECISION
CONTINUATION

PARSING OF DECISION TABLES	670803
DEVELOPMENT OF A COMPUTER PROGRAM FOR GENERATING TROUBLE-SHOOTING DECISION TREES	670901
CONVERSION OF LIMITED-ENTRY DECISION TABLES TO OPTIMAL COMPUTER PROGRAMS II: MINIMUM STORAGE REQUIREMENT	671102
SIMULATION WITH DECISION TABLES	680101
AN INTRODUCTION TO DECISION TABLES	680701
THE USE OF DECISION TABLES WITHIN SYSTEMATICS	680902
GOOD DECISION TABLES AND THEIR USES	680901
AMBIGUITY IN LIMITED ENTRY DECISION TABLES	681001
THE INTERPRETATION OF LIMITED ENTRY DECISION TABLE FORMAT AND RELATIONSHIPS AMONG CONDITIONS	691101
ON THE CONVERSION OF DECISION TABLES TO COMPUTER PROGRAMS	700601
DECISION TABLE TRANSLATION	700902
CONVERSION OF LIMITED-ENTRY DECISION TABLES TO COMPUTE PROGRAMS A PROPOSED MODIFICATION TO POLLACKS ALGORITHM	710201

DEFINING

DEFINING ADEQUATE SYSTEMS DOCUMENTATION	701201

DESIGN

THE DESIGN OF VISUAL DISPLAYS	700901
MAKING AUTOMATED COMPUTER PROGRAM DOCUMENTATION A FEATURE OF TOTAL SYSTEM DESIGN	701103
AUTOMATED ENGINEERING DESIGN AN APPROACH TO AUTOMATED DOCUMENTATION	701108

DEVELOPMENT

DEVELOPMENT OF A COMPUTER PROGRAM FOR GENERATING TROUBLE-SHOOTING DECISION TREES	670901

FORTRAN
CONTINUATION

SYSTEMS FORTRAN REFERENCE MANUAL PUBLICATION NUMBER 60174900 — 691103

IBM SYSTEM/360 OPERATING SYSTEM FORTRAN IV (G AND H) PROGRAMMERS GUIDE PROGRAM NUMBERS 360S-FO-50C 360S-FO-520 — 700603

FORTRAN VARIABLE NAME DOCUMENTER — 720607

FRESS

FRESS USER'S GUIDE — 720617

FULL

IBM SYSTEM/360 OPERATING SYSTEM FULL AMERICAN NATIONAL STANDARD COBOL PROGRAMMERS GUIDE NUMBERS 360S-CB-545 — 700602

GENERATING

DEVELOPMENT OF A COMPUTER PROGRAM FOR GENERATING TROUBLE-SHOOTING DECISION TREES — 670901

AN AUTOMATED SYSTEM FOR GENERATING PROGRAM DOCUMENTATION — 701106

GENERATION

NMCS INFORMATION PROCESSING SYSTEM IBM 1410/7010 GENERAL PURPOSE COMPONENTS MULTIFILE OUTPUT AND FILE GENERATION — 680302

A TECHNIQUE FOR COMPUTER FLOWCHART GENERATION. — 680801

GOVERNMENT

COMPUTERIZING A GOVERNMENT DATA SYSTEM: A MANAGEMENT OVERVIEW OF THE STEPS REQUIRED AND THE TIME NEEDED. — 671202

GRAPHICS

THE GROWTH OF COMPUTER GRAPHICS AT BELL LABORATORIES. — 680601

A SURVEY OF DATA STRUCTURES FOR COMPUTER GRAPHICS SYSTEMS — 710303

ARPA NETWORK SERIES I.INTRODUCTION TO THE ARPA NETWORK AT RAND AND TO THE RAND VIDEO GRAPHICS SYSTEM — 710901

I-IV

EVOLUTIONARY SYSTEM FOR DATA PROCESSING, VOLUMES I-IV. — 000102

IBM

PROGRAM ENABLES IBM 7070/7074 TO DRAW FLOWCHARTS. — 631001

USER'S MANUAL NMCS INFORMATION PROCESSING SYSTEM (NIPS) (IBM 1410/7010) — 680102

NMCS INFORMATION PROCESSING SYSTEM IBM 1410/7010 GENERAL PURPOSE COMPONENTS MULTIFILE OUTPUT AND FILE GENERATION — 680302

IBM SYSTEM/360 OPERATING SYSTEM LINKAGE EDITOR AND LOADER PROGRAM NUMBERS 360S-ED-510 360S-ED-521 360S-LD-547 — 691102

IBM SYSTEM/360 OPERATING SYSTEM FULL AMERICAN NATIONAL STANDARD COBOL PROGRAMMERS GUIDE NUMBERS 360S-CB-545 — 700602

IBM SYSTEM/360 OPERATING SYSTEM FORTRAN IV (G AND H) PROGRAMMERS GUIDE PROGRAM NUMBERS 360S-FO-500 360S-FO-520 — 700603

IBM SYSTEM/360 OPERATING SYSTEM ASSEMBLER F PROGRAMMERS GUIDE PROGRAM NUMBER 360L-AS-037 — 701203

IBM SYSTEM/360 OPERATING SYSTEM PL/I(P) PROGRAMMERS GUIDE PROGRAM NUMBER 360S-NL-511 — 710101

ILLIAC

ILLIAC IV QUARTERLY PROGRESS REPORT — 671201

INDEX

INDEX PREPARATION FOR PUBLICATIONS INDUSTRY — 680902

INDUSTRY

INDEX PREPARATION FOR PUBLICATIONS INDUSTRY — 680902

INFORMATION

INFORMATION SCIENCES: SOME RESEARCH DIRECTIVES — 660401

INFORMATION
CONTINUATION

USER'S MANUAL NMCS INFORMATION PROCESSING SYSTEM (NIPS) (IBM 1410/7010) 680102

BIBLIOGRAPHY ON INFORMATION SCIENCES. VOLUME I 680201

NMCS INFORMATION PROCESSING SYSTEM IBM 1410/7010 GENERAL PURPOSE COMPONENTS MULTIFILE OUTPUT AND FILE GENERATION 680302

COMPUTER IN INFORMATION SCIENCES. VOLUME III 681002

ON-LINE MANAGEMENT SYSTEMS INFORMATION 690401

AMERICAN NATIONAL STANDARD FLOWCHART SYMBOLS AND THEIR USAGE IN INFORMATION PROCESSING 700905

NETWORK INFORMATION CENTER USER GUIDE 720701

INQUIRY

SLIP - SOURCE LIBRARY INQUIRY PROGRAM, NO. 360D-00.3.009 660601

INTEGRATED

COST ADVANTAGES OF AN INTEGRATED DOCUMENTATION APPROACH 701113

INTEGRATION

THE INTEGRATION OF SYSTEM SPECIFICATIONS AND PROGRAM COSING. A REPORT ON EXPERIENCE OF THE MEDLARS II PROJECT. 701114

INTERIM

AN INTERIM PROGRESS REPORT OF COMPUTER OUTPUT MICROFILM ACTIVITIES AND EXPERIENCES 700701

INTERPRETATION

THE INTERPRETATION OF LIMITED ENTRY DECISION TABLE FORMAT AND RELATIONSHIPS AMONG CONDITIONS 691101

INTRODUCTION

AN INTRODUCTION TO DECISION TABLES 680701

INTRODUCTION
CONTINUATION

PROBLEM LANGUAGE ANALYZER (PLAN) USERS' INTRODUCTION 690101

ARPA NETWORK SERIES I.INTRODUCTION TO THE ARPA NETWORK AT RAND AND TO THE RAND VIDEO GRAPHICS SYSTEM 710901

JOSS

THE JOSS PRIMER 670804

KRONOS

EDIT - KRONOS TEXT EDITOR (EDIT) REFERENCE MANUAL 710104

KRONOS TEXT EDITOR (EDIT) 710105

LABORATORIES

THE GROWTH OF COMPUTER GRAPHICS AT BELL LABORATORIES. 680601

LANGUAGE

PROBLEM LANGUAGE ANALYZER (PLAN) USERS' INTRODUCTION 690101

XEROX TEXT LANGUAGE AND OPERATIONS REFERENCE MANUAL 720503

LARGE

DATA COMPRESSION IN LARGE BUSINESS-ORIENTED FILES 711002

LIBRARIAN

LIBRARIAN 720601

LIBRARY

SLIP - SOURCE LIBRARY INQUIRY PROGRAM, NO. 3600-00.3.009 660601

S/360 ASSEMBLER/COMPILER LISTINGS LIBRARY MAINTENANCE, NO. 3600-00.0.015 681101

CLOT - CARD LIBRARY ON TAPE 700604

A LIBRARY OF DATA COLLECTION AND MANIPULATION SUBROUTINES 710902

OPERATIONS

XEROX TEXT LANGUAGE AND OPERATIONS REFERENCE MANUAL — 720503

OPERATOR'S

SYSTEM/360 FLOWCHART USER'S MANUAL OPERATOR'S MANUAL — 690102

OPTIMAL

CONVERSION OF LIMITED-ENTRY DECISION TABLES TO OPTIMAL COMPUTER PROGRAMS I: MINIMUM AVERAGE PROCESSING TIME — 660801

CONVERSION OF LIMITED-ENTRY DECISION TABLES TO OPTIMAL COMPUTER PROGRAMS II: MINIMUM STORAGE REQUIREMENT — 671102

OUTPUT

NMCS INFORMATION PROCESSING SYSTEM IBM 1410/7010 GENERAL PURPOSE COMPONENTS MULTIFILE OUTPUT AND FILE GENERATION — 680302

AN INTERIM PROGRESS REPORT OF COMPUTER OUTPUT MICROFILM ACTIVITIES AND EXPERIENCES — 700701

OVERLAY

OVERLAY TREE PROCESSOR — 700101

OVERVIEW

COMPUTERIZING A GOVERNMENT DATA SYSTEM: A MANAGEMENT OVERVIEW OF THE STEPS REQUIRED AND THE TIME NEEDED. — 671202

PACKAGES

PERSPECTIVE ON FLOWCHART PACKAGES — 710301

PANVALET

PANVALET - THE DIRECT ACCESS LIBRARY SYSTEM — 720502

PARSING

PARSING OF DECISION TABLES — 670803

PERSPECTIVE

PERSPECTIVE ON FLOWCHART PACKAGES — 710301

PERT

A PROGRAM FOR THE CONSTRUCTION OF PERT FLOWCHARTS — 671101

PLOT

AUTOMATIC DATA PROCESSING SUPPORT FOR FLAG PLOT — 660402

PLOTTING

AUTOMATED PLOTTING OF FLOWCHARTS ON A SMALL COMPUTER — 650101

PLUS

PLUS D/A, A DIRECT ACCESS SOFTWARE SECURITY AND CONTROL SYSTEM — 720201

PL360

SYNTAX-DIRECTED DOCUMENTATION FOR PL360 — 700401

POLLACKS

CONVERSION OF LIMITED-ENTRY DECISION TABLES TO COMPUTE PROGRAMS A PROPOSED MODIFICATION TO POLLACKS ALGORITHM — 710201

POST

STUDY OF MACHINE-AIDED POST EDITING — 680202

PREPARATION

AUTOMATIC PREPARATION OF FLOWCHART LISTING. — 5[illegible]0101

INDEX PREPARATION FOR PUBLICATIONS INDUSTRY — 680902

RUNOFF: A PROGRAM FOR THE PREPARATION OF DOCUMENTS — 700301

PRIMER

THE JOSS PRIMER — 670804

PROCESSING

EVOLUTIONARY SYSTEM FOR DATA PROCESSING, VOLUMES I-IV. — [illegible]0102

AUTOMATIC DATA PROCESSING SUPPORT FOR FLAG PLOT — 660402

QUICK-DRAW

QUICK-DRAW 720611

RAND

ARPA NETWORK SERIES I.INTRODUCTION TO THE ARPA NETWORK AT RAND AND TO THE RAND VIDEO GRAPHICS SYSTEM 710901

REFERENCE

CONTROL DATA 6000 SERIES COMPUTER SYSTEMS REFERENCE MANUAL NUMBER 60100000 650701

CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COBOL REFERENCE MANUAL PUBLICATION NUMBER 60191200 690901

CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COMPASS REFERENCE MANUAL PUBLICATION NUMBER 60190900 691001

CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS FORTRAN REFERENCE MANUAL PUBLICATION NUMBER 60174900 691103

ED PROCESSOR - UNIVAC 1100 SERIES OPERATING SYSTEM PROGRAMMER REFERENCE 710103

EDIT - KRONOS TEXT EDITOR (EDIT) REFERENCE MANUAL 710104

TECO - TEXT EDITOR AND CORRECTOR - REFERENCE MANUAL 710603

XEROX TEXT LANGUAGE AND OPERATIONS REFERENCE MANUAL 720503

RELATIONSHIPS

THE INTERPRETATION OF LIMITED ENTRY DECISION TABLE FORMAT AND RELATIONSHIPS AMONG CONDITIONS 691101

REPORT

ILLIAC IV QUARTERLY PROGRESS REPORT 671201

AN INTERIM PROGRESS REPORT OF COMPUTER OUTPUT MICROFILM ACTIVITIES AND EXPERIENCES 700701

THE INTEGRATION OF SYSTEM SPECIFICATIONS AND PROGRAM COSING. A REPORT ON EXPERIENCE OF THE MEDLARS II PROJECT. 701114

REQUIRED

COMPUTERIZING A GOVERNMENT DATA SYSTEM: A MANAGEMENT OVERVIEW OF THE STEPS REQUIRED AND THE TIME NEEDED. 671202

REQUIREMENT

CONVERSION OF LIMITED-ENTRY DECISION TABLES TO OPTIMAL COMPUTER PROGRAMS II: MINIMUM STORAGE REQUIREMENT 671102

RESEARCH

INFORMATION SCIENCES: SOME RESEARCH DIRECTIVES 660401

RESEARCH IN COMPUTER-ASSISTED DOCUMENTATION OF COMPUTER PROGRAMS 701202

RETRIEVAL

MICROFICHE STORAGE AND RETRIEVAL SYSTEMS STUDY: 700801

RULE

CONVERSION OF DECISION TABLES TO COMPUTER PROGRAMS BY RULE MASK TECHNIQUES 661101

RUNOFF

RUNOFF: A PROGRAM FOR THE PREPARATION OF DOCUMENTS 700301

SCIENCES

INFORMATION SCIENCES: SOME RESEARCH DIRECTIVES 660401

BIBLIOGRAPHY ON INFORMATION SCIENCES. VOLUME I 680201

COMPUTER IN INFORMATION SCIENCES. VOLUME III 681002

STANDARDS GUIDE FOR SPACE AND EARTH SCIENCES COMPUTER SOFTWARE 720103

SCIENTIFIC

AUTOFLOW ENHANCEMENTS OF DOCUMENTATION AND MAINTENANCE OF SCIENTIFIC APPLICATIONS 701101

SYSTEM
CONTINUATION

DOCUMENTATION 701106

THE INTEGRATION OF SYSTEM SPECIFICATIONS AND PROGRAM CODING. A REPORT ON EXPERIENCE OF THE MEDLARS II PROJECT. 701114

IBM SYSTEM/360 OPERATING SYSTEM ASSEMBLER F PROGRAMMERS GUIDE PROGRAM NUMBER 360L-AS-037 701203

IBM SYSTEM/360 OPERATING SYSTEM PL/I(F) PROGRAMMERS GUIDE PROGRAM NUMBER 360S-NL-511 710101

UNIVAC 1100 SERIES OPERATING SYSTEM 710102

ED PROCESSOR - UNIVAC 1100 SERIES OPERATING SYSTEM PROGRAMMER REFERENCE 710103

MAINTENANCE AND ENHANCEMENT OF THE AED SYSTEM 710602

ARPA NETWORK SERIES I.INTRODUCTION TO THE ARPA NETWORK AT RAND AND TO THE RAND VIDEO GRAPHICS SYSTEM 710901

CFMS - CARD FILE MAINTENANCE SYSTEM 720101

PLUS D/A, A DIRECT ACCESS SOFTWARE SECURITY AND CONTROL SYSTEM 720201

PANVALET - THE DIRECT ACCESS LIBRARY SYSTEM 720502

SYSTEM--OS

SYSTEM/360 ADMINISTRATIVE TERMINAL SYSTEM--OS 680303

SYSTEMATICS

THE USE OF DECISION TABLES WITHIN SYSTEMATICS 680802

SYSTEMS

CONTROL DATA 6000 SERIES COMPUTER SYSTEMS REFERENCE MANUAL NUMBER 60100000 650701

DECISION TABLES FOR BERRER MANAGEMENT SYSTEMS 660301

ON-LINE MANAGEMENT SYSTEMS INFORMATION 690401

SYSTEMS
CONTINUATION

CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COBOL REFERENCE MANUAL PUBLICATION NUMBER 60191200 690901

CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS COMPASS REFERENCE MANUAL PUBLICATION NUMBER 60190900 691001

CONTROL DATA 6400/6500/6600 COMPUTER SYSTEMS FORTRAN REFERENCE MANUAL PUBLICATION NUMBER 60174900 691103

MICROFICHE STORAGE AND RETRIEVAL SYSTEMS STUDY: 700801

DEFINING ADEQUATE SYSTEMS DOCUMENTATION 701201

A SURVEY OF DATA STRUCTURES FOR COMPUTER GRAPHICS SYSTEMS 710303

TABLE

THE INTERPRETATION OF LIMITED ENTRY DECISION TABLE FORMAT AND RELATIONSHIPS AMONG CONDITIONS 691101

DECISION TABLE TRANSLATION 700902

TABLES

LOGIC STRUCTURE TABLES 610601

TABLES, FLOW CHARTS, AND PROGRAM LOGIC 620901

USING DECISION STRUCTURE TABLES 640301

USE OF DECISION TABLES IN COMPUTER PROGRAMMING 650102

CONVERSION OF LIMITED ENTRY DECISION TABLES TO COMPUTER PROGRAMS 651101

DATA, DOCUMENTATION, AND DECISION TABLES 660101

PROGRAMMING DECISION TABLES IN FORTRAN, COBOL OR ALGOL 660102

DECISION TABLES FOR BERRER MANAGEMENT SYSTEMS 660301

CONVERSION OF LIMITED-ENTRY DECISION TABLES TO OPTIMAL COMPUTER PROGRAMS II MINIMUM AVERAGE PROCESSING TIME 660801

APPENDIX B

EXISTING DOCUMENTATION AIDS

AUTOMATIC SYSTEM FOR COMPUTER PROGRAM DOCUMENTATION

Working Paper No. 1

July 25, 1972

Proprietary Documentation Systems

by

Randy Birge

Texas A&M University

Texas Engineering Experiment Station

B-1a

ABSTRACT

This report is intended to acquaint the reader with features supplied by proprietary automatic documentation software packages currently on the market. A number of systems are listed with their corresponding purchase and rental costs, description of outputs, source languages processed, system environment, and in some cases sample outputs of the system.

Table of Contents

INTRODUCTION

Every data processing manager and programmer is familiar with the problem posed by the requirements for accurate program documentation. The preparation of this documentation has traditionally been time consuming and, consequently, it has often been set aside in order to allow the programmer to work on higher priority projects.

The major goal inherent in any good documentation procedure is to construct a system for communicating a document's content to a human being. A well documented computer program enables user personnel to understand the program's objectives, its relationship to other programs in the overall installation, and its position in the system workflow.

To relieve the programmer of much of the responsibility of program documentation many proprietary automatic documentation system have developed over the past few years. Many of these are generalized documentation packages which can accept source code as input and generate routines for computer analysis of the program requiring documentation as well as routines for producing reports on graphical representation that describe the program.

To achieve a permanent record of the details of an existing program, the following program feature require documenting:

1) Functional relation
2) Cross-reference
3) Furnish suitable cross-referencing facilities so that references to and from a given data element can be adequately presented.

Each cross-reference can be indicated on the flowchart at the place where the related element is mentioned, or a separate listing can be printed. A separate list has the advantage of furnishing all cross-references in a convenient format.

4) Allow flexibility in the level of detail. A chart can be developed on a symbol-per-statement basis, or it can be presented as an overall picture of the program's functions.
5) Convert comments in the source statements as well as other explanatory remarks into annotative flowchart statements.
6) In the absence of explanatory remarks, generate the necessary annotations.
7) Generate symbols that conform to accepted conventions or adhere to documented standards such as the ANSI standards.

Some of the proprietary automatic documentation packages listed in this report contain all of the above features and more, while some simply give simple cross-references. The purchase and rental costs of these documentation packages are a function of the number of output features they list. Purchase price range from $175 for a simple cross-referencing documentation package, FXREF, to $7,000 for the AUTOFLOW system which contains possibly the greatest number of output features.

The information for this report was obtained from the various software companies which produce proprietary automatic documentation software packages and from the Auerbach Computer Technology Reports. Letters requesting information on particular software packages were sent out to thirty five software companies. Of these original thirty five requests, about fifty percent of the companies responded.

A second letter was sent out to those companies who did not respond to the first letter. Less than fifty percent of these companies responded to the second letter.

Some of the information which could not be obtained from the software companies which produced these automatic documentation packages was supplemented by the Auerbach Computer Technology Reports.

COMPARISON CHARTS

COMPARISON CHARTS

Introduction

The following charts present many characteristics of existing documentation packages. These charts are intended as a guidance for comparing available documentation packages. Quick comparisons may be made through the use of the charts, and further information on a particular documentation package can then be found in the Package Reports section of this report.

The charts are divided into two major sections:

System Requirements

Characteristics

	FEATURE \ PACKAGE	AUTOCHART Beloit Computer Center	AUTODIAGRAMMER II Aires Corp.	AUTODOC Data Instruments Co.
SYSTEM REQUIREMENTS	System	IBM 360/25	IBM 360/30 , UNIVAC 9300	any capable of COBOL support
	Main Storage	32K	52K	28-64K
	Auxilary Storage	one disk (2311 or 2314	Disc	Magnetic tape or disc
	Input/Output	card reader/ punch, printer	Disc, Cardreader, Printer	line printer,card reader disc, tape
	Operating System	DOS,OS-MVT	DOS, OS	any capable of COBOL support
	Source Language	(not available)	Assembly Language	COBOL and IBM assembly
CHARACTERISTICS	Package Type	Program Documentation detailed flowchart, cross-reference listing of labels	Program Documentation; detail flowchart,or high level. cross-reference lists, logic charts, label cross-reference, unmatched label tables, record and report layout source listing	Documentation Aid; cover page, source list, error list, report and record lay-out, cross-reference list, detail flowchart, brief logic chart
	Logic Flow Presentation	Detail flowchart at same level as source code	Detail flowchart at source code level; variable symbols contain source codes, flow is vertical and page formatting is available. high-level logic chart ill.par.relationships	narrative description from source comments, detail flowchart with ANSI or IBM symbols same level of detail as source code, depicts page, connector no. of referenced statements and referencing statements. logic chart depicts statements that affect only logic flow
	Cross References	shows location of each label in the flowchart, location of each label that branches to that statement	Paragraph cross-references in high-level logic chart ; Label cross-reference table, Altered Statements Unmatched unreferenced labels	listing of data items, procedure names,ext. names,literals,figurative constants,system names alphanumeric ordering gives label type,defining and referencing source code number also on detail flowchart
	Man-Machine Interface	Input-card reader Output-printer	Input-card reader Output- printer	Input-cards,disc,tape; can input up to 99 programs in batched made from card reader or from system library Output-printer
	Cost	Purchase - $950.00 Rental - (none)	Purchase - $3,200 Rental - (none)	Purchase - $4,800 (3-yr lease) Rental - (none)
	Source Programs Processer	COBOL	COBOL FORTRAN BAL } optimal	COBOL IBM assembly
	Comments	Two phase system; flow-charter, cross-referencing	about 20-30 installations	about 50 installations

FEATURE \ PACKAGE		AUTOFLOW (1964) Applied Data Research	COMCHART Compress	DYNACHART Applications Programming Co.
SYSTEM REQUIREMENTS	System	IBM 360,1400;RCA Spectra 70;H200	IBM 360/370; RCA Spectra 70	any with COBOL Compiler
	Main Storage	Support for OS,DOS,TOS, TDOS	65K bytes	30K character
	Auxilary Storage	System residence device	4 sequential files	1 disc or 4 tapes
	Input/Output	card reader,tape,disc, line printer	card reader,tape,disc, line printer	I/O - card reader, printer disc,tape
	Operating System	OS,DOS,TOS(IBM 360), TDOS(RCA)	OS,TOS,TDOS	any with cobol compiler
	Source Language	Basic Assembly Language	COBOL and Assembly	COBOL
CHARACTERISTICS	Package Type	Documentation aid; generates a flowchart used for debugging summarizing an existing system constructing a preimplementation flowchart	Documentation aid;produces a flowchart,deck listing,a cross-reference index of element names;also a design language for developing programs	Documentation aid which produces;flowchart, diagnostics,cross-references
	Logic Flow Presentation	In form of text and flow-charts; flowchart at same level as source code, distinct units for processing blocks, page,source card,symbol (box) nos. on and off-page connectors	Standard symbols used in flowchart;built in cross-references;detailed analysis of each statement	Flowcharts at same level of detail as source code;separate page per subroutine; paragraph not split between logical pages
	Cross References	Table of contents and references index preceeds flowchart and contains cross-referencing information Listing of COBOL data names in alphanumeric order with page,flow-chart box #,and source sequence #.	alphabetic index of all elements outline of procedures, listing all incoming and outgoing references	Labels alphabetically listed as are defining source code sequence number and all statements referencing label
	Man-Machine Interface	Input - card reader Output - printer	Input - card reader, tape disc Output - line printer	Input - card,tape,disc Output - printer
	Cost	Purchase - $3,000-7,000 Rental - per installation basis	Purchase - (none) Rental - $2,425 (1 yr); $4,225 (10 yrs)	Purchase - $4,400 (1st yr) Rental - $700/yr after 1st yr
	Source Programs Processer	COBOL PL/I Assembly FORTRAN	COBOL, Assemble, Design Language	COBOL
	Comments	About 1400 installations	Over 50 installations	About 45 installations

	FEATURE \ PACKAGE	EZFLOW Systonetics Corp.	FACTS Bonner & Moore Asso.,Inc.	FLOWGEN/F Calcomp
SYSTEM REQUIREMENTS	System	IBM 360,CDC 6000 Series	Sigma 7	Not Available
	Main Storage	110K bytes (IBM), 32K words (CDC)	Not Available	Not Available
	Auxilary Storage	none	Not Available	Not Available.
	Input/Output	card reader,tape, or disc, line printer	Card reader,tape,printer tape	card reader, CalComp plotter
	Operating System	OS(IBM),SCOPE(CDC)	Not Available	Not Available
	Source Language	FORTRAN IV	Not Available	Not Available
CHARACTERISTICS	Package Type	Documentation Aid that generates flowchart input/output source deck a cross-reference list statement reference table	Documentation aid; cross referencing	Program Documentation Detailed Flowchart
	Logic Flow Presentation	flowchart at same level as source code; up and down single-col flow;	None	Detail Flowchart
	Cross References	Listing of statement labels and references to show old and resequenced source code label numbers and source list line number	Six reports are given; common report,local report,format statement statement label report, recap, global report	(none)
	Man-Machine Interface	Input-card reader,tape, or disc Output-printer	Input-card reader,tape Output-printer,tape	Input-card reader,drum, disc, or tape Output-CalComp Plotter
	Cost	Purchase - $1,750 Rental -(none)	Purchase-Not Available Rental - Not Available	Not available
	Source Programs Processer	FORTRAN	FORTRAN	FORTRAN
	Comments	under 10 installations	Not Available	Not Available

	FEATURE \ PACKAGE	FORFLO DNA Systems, Inc.	FORTRAN VARIABLE NAME DOCUMENTER Data for Management Decision	QUICK-DRAW National Computer Analysts
SYSTEM REQUIREMENTS	System	IBM 1130/1800	Designed for user's configuration	Any with COBOL compiler
	Main Storage	1130(8K core); 1800 (10K variable core)	Capable of supporting FORTRAN compiler	32K bytes(DOS,TOS);65K bytes(OS
	Auxilary Storage	1 disc	1 disc	System residence device
	Input/Output	disc,card reader, line printer	Card reader,line printer	card reader,tape,disc, line printer
	Operating System	Monitor, TSX version, MTX	Specific system not required	any with COBOL compiler
	Source Language	FORTRAN and Assembler Subroutines	FORTRAN	COBOL, Assembler
CHARACTERISTICS	Package Type	Documentation package; gives flowcharts and resequenced source listing	Documentation tool;source list and alphabetical list of referenced variable names	Documentation aid source programs into flowcharts elates cross-references diagnostic checklist, modifies instruction list available for de-bugging
	Logic Flow Presentation	Flowchart at same level as source code, source statements within blocks	(none)	Flowchart at same level as source code uses symbols,text,flow-connectors,reference information each element of chart identified by page, block,QUICK-DRAW-Assigned source card sequence, & statement nos.or tag double-page format
	Cross References	(none)	Sequence number of all statements that reference each variable name	Referencing described by flowchart coordinates & source code no. Cross-reference tables include data-defined fields Alphabetic listing of tag names procedural statements and all sub-routines data and procedure names procedural logic
	Man-Machine Interface	Input-cards or disc Output-printer or disc	Input-card reader Output-line printer	Input-card reader,tape, disc Output-printer
	Cost	Purchase - $480(cards); $600(disc)	Purchase-$350(object); $500(source)	Purchase - none Rental - $1,400-3,100 (1yr); $1,900-6,300 (3yrs)
	Source Programs Processer	FORTRAN	FORTRAN	COBOL FORTRAN Assembly
	Comments	Resequence listing arranges source state-ment label nos. by fives.about 63 install.	About 6 installations	QUICK-DRAW is being used at over 400 installations

	FEATURE / PACKAGE	SUPEREF Mantech Corporation	FORDOC J. Toellner & Asso.	
SYSTEM REQUIREMENTS	System	CDC 6000 series	not Available	
	Main Storage	Not Available	Not Available	
	Auxilary Storage	Not Available	Not Available	
	Input/Output	Card reader,printer	Card reader/punch,printer	
	Operating System	Not Available	Not Available	
	Source Language	Not Available	Not Available	
CHARACTERISTICS	Package Type	Documentation Aid Cross-reference listing	Program restructuring & documentation Cross-reference listing and restructured source deck.	
	Logic Flow Presentation	(none)	(none)	
	Cross References	Produces a comprehensive symbolic name Cross-reference dictionary	Variable name cross-reference.	
	Man-Machine Interface	Not Available	Input - cardreader Output - printer, card punch	
	Cost	Not Available	Not Available - Purchase Not Available - Retail	
	Source Programs Processer	FORTRAN	FORTRAN	
	Comments	Not Available	Not Available	

PACKAGE REPORTS

PACKAGE REPORTS

Introduction

This section contains information on documentation aids packages currently available to the data processing market. Sample outputs are given for a number of the packages.

Each report has two major sections:

GENERAL - In this section features such as package purpose, system requirements, pricing, and source languages processed are given.

PACKAGE OUTPUT - This section lists the various outputs that the particular package gives. A description of each output is also given.

AUTOCHART

Beloit Computer Center, Inc.

GENERAL

AUTOCHART accepts COBOL source programs as input and produces a detailed flowchart and a cross-reference listing of all labels used in the program. These two phases can be run in one jobstream or they can be executed as separate programs.

The minimum configuration for using AUTOCHART is an IBM System 360 Model 25 on up with a 32K central processor, one disk storage file (2311 or 2314), one card reader/punch, and a line printer. A magnetic tape drive is optional if you wish to execute the two phases in one jobstream. The system operates either under DOS or OS-MVT.

AUTOCHART is a low cost system with limited capabilities. The purchase price for AUTOCHART is $950.00 complete.

PACKAGE OUTPUT

Flowchart - A detailed flowchart is given.

Cross-Reference - This listing shows the location of each label in the flowchart as well as the location of each label that branches to that statement.

(No output examples are available)

<u>AUTODIAGRAMMER II</u>

AIRES CORPORATION

GENERAL

AUTODIAGRAMMER II is used as a documentation aid and as a debug tool for COBOL source programs. Standard outputs are a detailed flowchart, a diagnostic listing, an altered statement cross-reference table, a record layout, a source listing, and an unmatched label table.

The package can be implemented on any IBM System/360 configuration capable of operating under DOS (Model 30 and up) with 64K bytes or OS (Model 40 and up) with 128K bytes. DOS requires a 2111 or 2314 disc; OS requires any direct access device. Additional requirements are one card reader and one printer. AUTODIAGRAMMER II is written in 360 assembly language.

Purchase price of the package is $3,200.

PACKAGE OUTPUT

The <u>detailed flowchart</u> is given in which each statement is displayed with a separate logic block. User can select one or two logical pages per physical page, margin size, and six or eight line-per-inch printing density. All symbols are of variable size. Processing comments associated with each source code are included within its corresponding flowchart symbol. Narrative source code comments are printed in-line and are not enclosed within symbols.

The <u>diagnostic table</u> lists all error the diagnostic error codes, the statements in error, the location of each statement in the detailed

C. 2

flowchart and the error code associated with each mistake.

The altered Statement Cross Reference listing displays the paragraph names and flowchart locations of the statements being altered, the location of each altering statement, and the name and flowchart location of each destination.

The high level logic chart displays the relationships between paragraphs in the program.

The input/output chart shows each file name, unit assigned, record name, and input or output relationship to the program.

The label cross-reference table displays each label by name, location, and sequence number and gives the flowchart location of each reference to the label by statements in the program.

The record layout graphically displays each record defined in the program with its field and sub-field name, length, and type specifications.

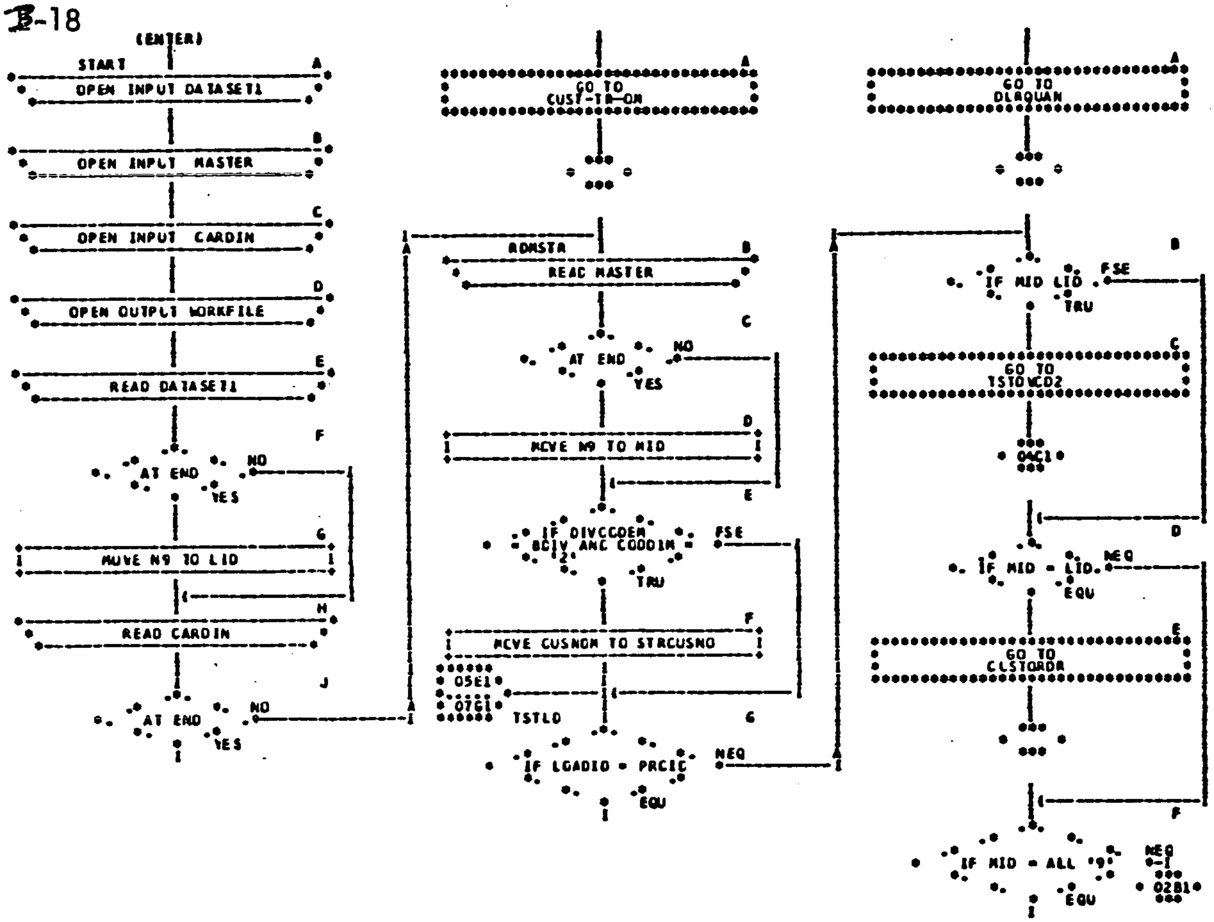
The sample report page displays a sample printed page that would result from the Report Definition Section of the user's program.

The source listing shows all the source statements and remarks as they appear in the source deck.

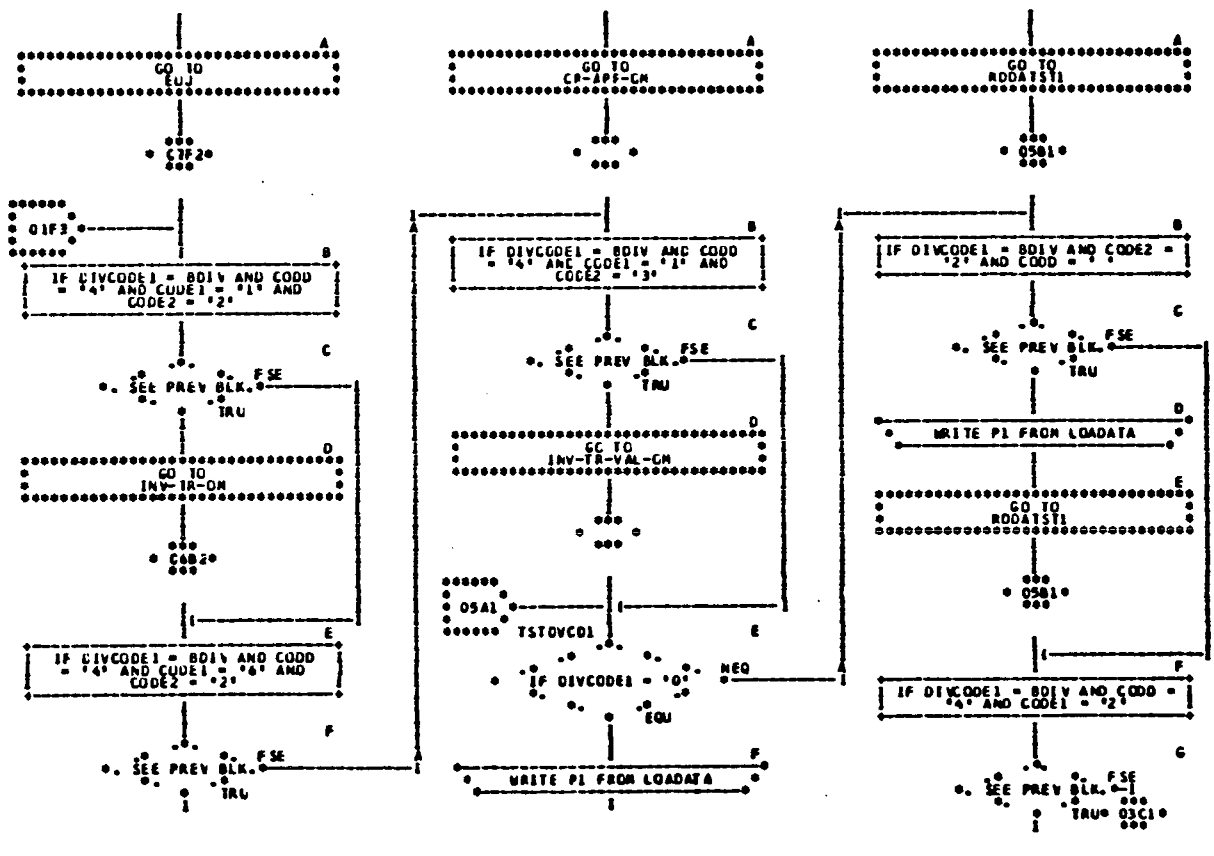
The unmatched label table displays each label which is defined but never referenced, and each label which is referenced but never defined.

06/05/69 ....• ARIES CORP.• CW SAMPLE • AUTODIAGRAMMER II •.......... PAGE 001

B-18



06/05/69 ....• ARIES CORP.• CW SAMPLE • AUTODIAGRAMMER II •.......... PAGE 002



```
                         COBOL DIAGNOSTIC ERROR CODES
A MISSING PROGRAM I.D. LABEL
B MISSING I.D. DIVISION LABEL
C MISSING ENVIRONMENT LABEL
D MISSING FILE CONTROL SECTION
E MISSING FD STATEMENT
F MISSING SELECT CLAUSE
G MISSING ASSIGN CLAUSE
H MISSING 01 LEVEL STATEMENT
I LABEL IN EXCESS OF THIRTY CHARACTERS
J INVALID PICTURE CLAUSE
K INVALID REDEFINES
L STATEMENT OUT OF SEQUENCE
M INVALID CONTINUATION OR MISSING PERIOD
N INVALID VALUE CLAUSE
R IMPROPER MARGIN USE
S INVALID SYNTAX
T PROCEDURE NAME IS A COBOL RESERVED WORD
W NON-TERMINATING SUBSCRIPT OR NON-NUMERIC LITERAL
```

```
          ..........* ARIES CORP.* ON SAMPLE * AUTODIAGRAMMER II  *..........
DIAGNOSTIC TABLE
*--------*-----------------------------------------------------------------*
| LOC    | SOURCE STATEMENT/INSTRUCTION                                    *--------*
|--------*-----------------------------------------------------------------| E.T.   |
|        | 000056        PCH MINV MOC MNR.                                 *--------*
|        | FF0C4L     C3  ERRORS3.                                        2 | M      |
|        | FF0050     C3  STRING  PICTURE IS X(8).                        2 | S      |
| 01B3   | JJJ120     IF  MAU . LID GO TO ISTCVLD2.             S709C3    2 | S      |
| C3F4   | ZZZ550 EOJ                                          S709038A    | S      |
| C7A3   |        MOVE.                                        S709C38A    | M      |
*--------*-----------------------------------------------------ERROR1----| T      |
                                                                         *--------*
```

Figure 4. Diagnostic Table and Error Codes

1. *High Level Logic Chart.*

The High Level Logic Chart illustrates the relationships between the paragraphs in a COBOL program. By portraying the linkage of process blocks, this chart assists the programmer in visualizing control transfer patterns and reviewing them for accuracy.

Each paragraph in the source program is shown with its name and flowchart location in a processing block. The processing blocks are listed in the order of appearance in the source program. They are connected by vertical lines whenever the preceding paragraph is *not* an EXIT or GO TO paragraph. To the left of each block are listed any paragraphs which transfer control into the block. To the right of each block are listed any paragraphs to which the block may transfer control. Figure 5. illustrates the format of the High Level Logic Chart.

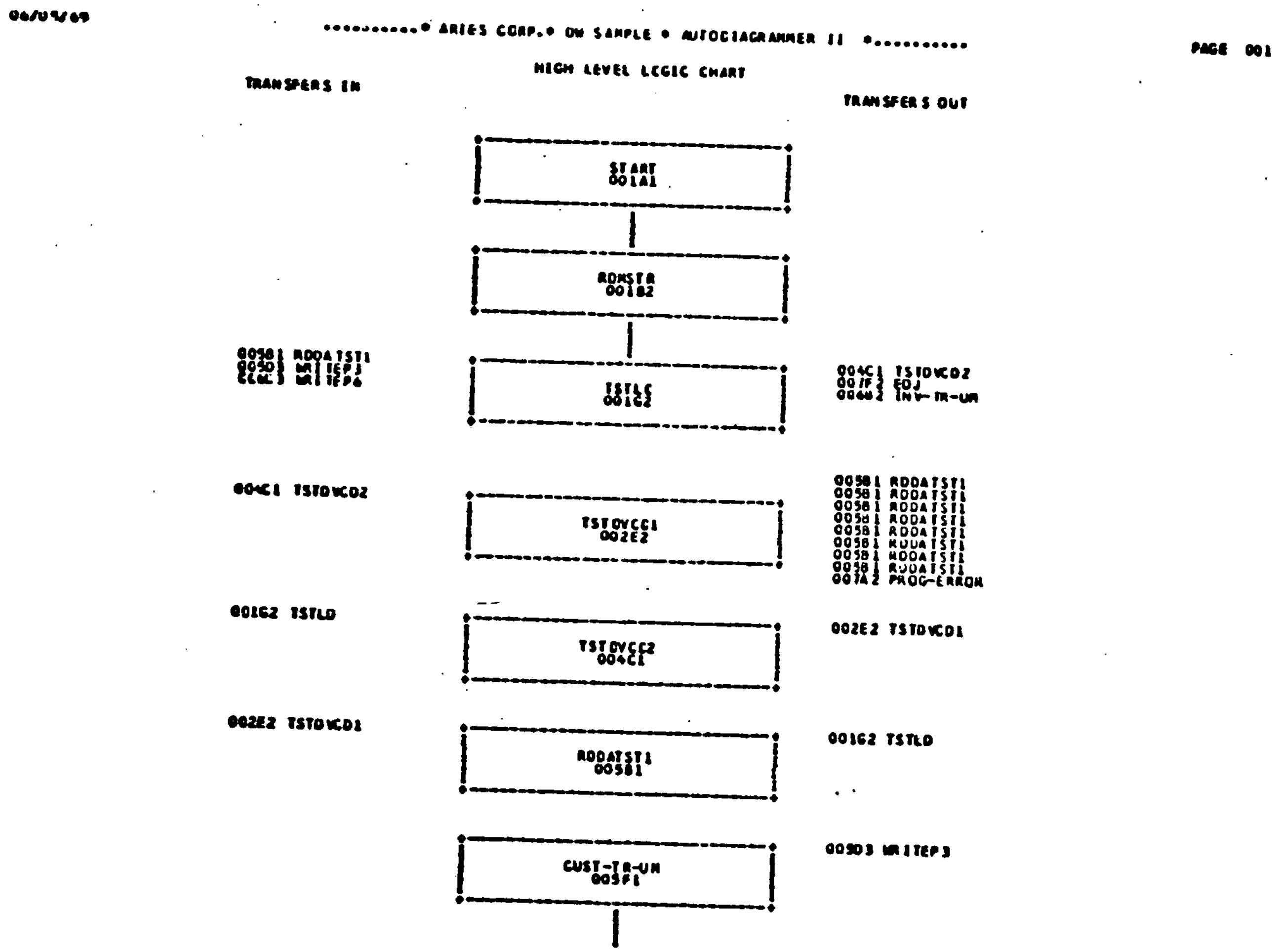
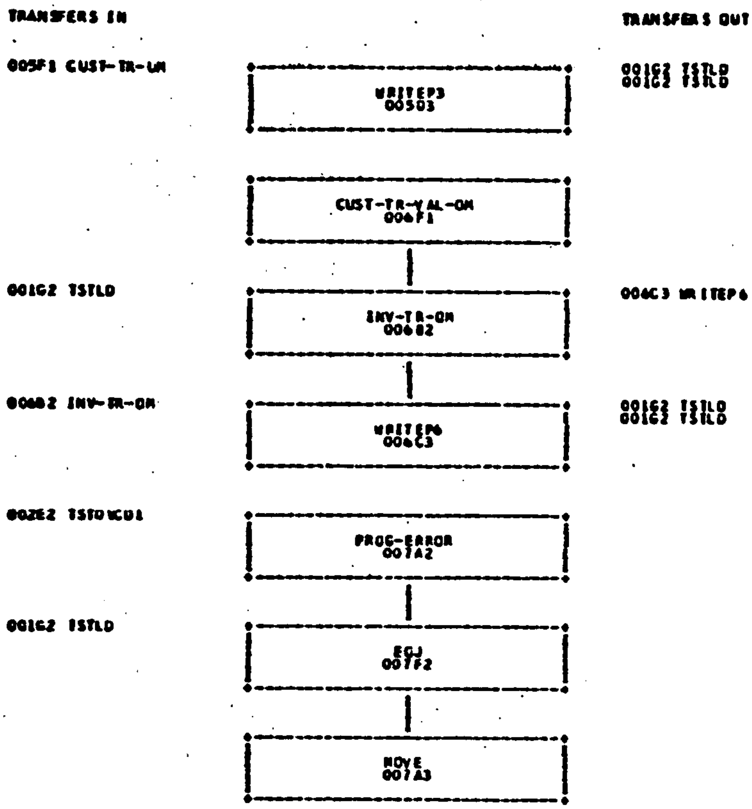


Figure 5. The High Level Logic Chart for the program in Appendix "A".

Continuation of Figure 5.

06/05/69 ....ARIES CORP.. DW SAMPLE . AUTODIAGRAMMER II .... PAGE 002

HIGH LEVEL LOGIC CHART



```
06/05/69          ..........* ARIES CORP.* ON SAMPLE * AUTODIAGRAMMER II *..........          PAGE 001

                                        I-O LOGIC CHART

                                          INPUT FILES
                  *----------------------------------------------------------------------*
                  | DATASET1I                                                            |
                  | LOADATA                                                              |
                  | ASSIGN TO 'SYS001' UTILITY                                           |
                  *----------------------------------------------------------------------*
                  *----------------------------------------------------------------------*
                  | MASTER                                                               |
                  | CUSREC                                                               |
                  | ASSIGN TO 'SYS002' UTILITY                                           |
                  *----------------------------------------------------------------------*

06/05/69          ..........* ARIES CORP.* ON SAMPLE * AUTODIAGRAMMER II *..........          PAGE 002

                                        I-O LOGIC CHART

                                    IDENTIFICATION DIVISION
                  *----------------------------------------------------------------------*
                  | PROGRAM-ID.  DEMONSTRATION                                           |
                  | AUTHOR. JOHN J PROGRAMMER.                                           |
                  | INSTALLATION. ANYWHERE USA.                                          |
                  | REMARKS. THIS PROGRAM EDITS TRANSACTIONS WHICH REQUIRE TWO           |
                  |         CUSTOMER RECORDS. PASS1 OF INTERNATIONAL ACCOUNTS            |
                  |         RECEIVABLE.                                                  |
                  |                                                                      |
                  *----------------------------------------------------------------------*

                                          OUTPUT FILES
                  *----------------------------------------------------------------------*
                  | WORKFILE                                                             |
                  | CUSDATA                                                              |
                  | ASSIGN TO 'SYS003' UTILITY                                           |
                  *----------------------------------------------------------------------*
```

Figure 6. The Input-Output Logic Chart.

Figure 7 shows the Label Cross-Reference Table produced by AUTODIAGRAMMER II using the source program in Appendix "A".

06/05/69 ....• ARIES CCRP.• ON SAMPLE • AUTODIAGRAMMER II •.......... PAGE 001

LABEL CROSS REFERENCE BY ALPHABETIC

LABEL	SEQ #	LOC	REFERENCE BY LOC
CUST-TR-ON	JJL050	005F1	
CUST-TR-VAL-ON	JJL180	006F1	
EOJ	ZZZ990	007F2	002A1
INV-TR-ON	JJM020	006B2	002C1
MOVE		007A3	
PROC-ERROR	ZZZ901	007A2	004B1
KOUATST1	JJL030	005B1	002A3,002E3,003B1,003F1,003D2,003A3,003D3,004A1
ROMSTR	JJJ085	001B2	
START	JJJ020	001A1	
TSTDWCD1	JJK010	002E2	005A1
TSTDWCD2	JJK150	004C1	001C3
TSTLO	JJJ110	001G2	005E1,006A1,006E1,007C1,007G1
WRITEP3	JJL120	005D3	005A3
WRITEP6	JJP040	006C3	006F2

Figure 7. Label Cross-Reference Table

The use of the Label Cross-Reference Table in checking transfer instructions is shown in Figure 7a. This kind of step by step analysis is applicable to labels which are referenced in the source program.

Contin .tion of Fig . 3.

06/05/69 ....• ARIES CORP.• DW SAMPLE • AUTODIAGRAMMER II •.......... PAGE 002

RECORD LAYOUT/S

CUSREC MASTER

LABEL	POSITION	CODE
CIVCODEM	1-1	AN
CUSNOM	2-9	
CUSM	2-7	AN
CUSAM	8-9	AN
CODD1M	10-10	AN
INVOICEM	11-20	AN
FILLEM	21-21	AN
CODE2M	22-22	AN
BATCHCATEM	23-28	AN
FRCUSM	29-36	AN
TOCUSM	37-44	AN
FRCTRM	45-50	AN
TOCTRM	51-56	AN
MAILM	57-57	AN
ICENIM	58-108	
CLASSM	58-58	AN
AREAM	59-61	AN
SPECDISCOUNTM	62-65	N
COUNTRYM	66-68	AN
DEALERM	69-71	AN
FILLER	72-88	AN
COUNTNAMEM	89-108	AN
CUSNAMEM	109-138	AN
LINE1M	139-168	AN
LINE2	169-198	AN
LINE3	199-228	AN
LINE4	229-258	AN
LINE5	259-288	AN
MAILCUSNAMEM	289-318	AN
LINEM1M	319-348	AN
LINEM2M	349-378	AN
LINEM3M	379-408	AN
LINEM4M	409-438	AN
LINEM5M	439-468	AN

06/05/69 ....• ARIES CORP.• DW SAMPLE • AUTODIAGRAMMER II •.......... PAGE 003

RECORD LAYOUT/S

CUSDATA WORKFILE

LABEL	POSITION	CODE
FILLER	1-1	AN
FILLER	2-9	AN
FILLER	10-10	AN
FILLER	11-21	AN
FILLER	22-22	AN
FILLER	23-28	N
FILLER	29-38	AN
FILLER	39-54	AN
CTACCCHGW	55-60	AN
FILLER	61-66	AN
MAILCODEW	67-67	AN
CUSTIDW	68-118	AN
CUSTNAMEW	119-148	AN
ADDR1W	149-178	AN
ADDR2W	179-208	AN
ADDR3W	209-238	AN
ADDR4W	239-268	AN
ADDR5W	269-298	AN
MAILNAMEW	299-328	AN
MADDR1W	329-358	AN
MADDR2W	359-388	AN
MADDR3W	389-418	AN
MADDR4W	419-448	AN
MADDR5W	449-478	AN

g. *Sample Report Page.*

When a program has been written using the COBOL Report Writer Feature, the programmer may request that AUTODIAGRAMMER II produce a sample printed page of the report. This output would show the report exactly as it is established in the Report Defin Section of the User's Program.

A sample report is given below.

```
MM/DD/YY                    ...........AUTODIAGRAMMER  II  SAMPLE  COBOL  FLOWCHART...........              PAGE 001
SAMPLE REPORT PAGE
                                               ********STANDARD HEADINGS********
                         AJAX MANUFACTURING COMPANY
                        QUARTERLY EXPENDITURES REPORT
                           AAAAAAAAAEXPENDITURES
MONTH      DAY    SECT  NO. OF ITEMS   TYPE        COST  CUMULATIVE COST

AAAAAAAAA  99     XXX          Z9          A       ZZ9.99

PURCHASES AND COST FORZ9-99-ZZ9                   $$$9.99        $$$$9.99
              TOTAL COST FOR AAAAAAAAAWAS   $$$9.99
              TOTAL COST FOR QUARTER WAS   $$$$9.99




AAAAAAAAA  99     XXX          Z9          A       ZZ9.99


                            AAAAAAAAACONTINUED
                                                  PAGE-99




MM/DD/YY                    ...........AUTODIAGRAMMER  II  SAMPLE  COBOL  FLOWCHART...........              PAGE 002
SAMPLE REPORT
                                               ********OVERFLOW HEADINGS********


                            AAAAAAAAACONTINUED
MONTH      DAY    SECT  NO. OF ITEMS  TYPE        COST  CUMULATIVE-COST


                                                  PAGE-99
                                END OF REPORT
```

A Sample Report produced by AUTODIAGRAMMER II.

06/05/69 ....* ARIES CORP.* ON SAMPLE * AUTODIAGRAMMER II *.......... PAGE 001

UNMATCHED LABEL TABLE

NEVER REFERENCED	LOC
START	001A1
NUMSIN	001B2
CUST-TR-CM	005F1
CUST-TR-VAL-CM	006F1
MOVE	007A3

NO MATCH	LOC
CUST-TR-CM	001A2
LARGUAN	001A3
CUSTCHCK	001E3
CR-APP-CM	002A2
INV-TR-VAL-CM	002D2
CUST-TR-NUM	004E1
CUST-TR-VAL-NUM	004C2
INV-TR-NUM	004F2
INV-TR-VAL-NUM	004C3
CR-APP-NUM	004F3
CALCINV	006A2

Figure 9. Unmatched Label Table.

04/27/70 TABLE OF DIAGNOSTICS AUTOFLOW CHART SET - SAMPLE PAGE 1

LOCATION		DIAGNOSTIC
CARD ID	PAGE/BOX	
012500	2.01	IMPROPER USE OF RESERVED WORD - START
019600	3.18	UNDEFINED - 'ERROR' EXTERNAL REFERENCE
027231	5.23	IMPROPER USE OF RESERVED WORD - NOTE
027231	5.23	NO ENTRANCE TO THIS STATEMENT
027232	5.23	UNDEFINED PROCEDURE REFERENCE - ERROR-1
027233	6.01	ALTERED LINE NOT GO TO - END-OF-JOB
027233	6.01	UNDEFINED PROCEDURE REFERENCE - ERROR-2
027233	6.01	NO ENTRANCE TO THIS STATEMENT
027234	6.02	INVALID SYNTAX - 2

AUTODOC

DATA INSTRUMENTS COMPANY

GENERAL

AUTODOC accepts COBOL source code and, when implemented on an IBM 360, will also accept assembly language source code. AUTODOC generages a cover page, a source program listing, and an error list. At the user's option it can also generate a document for describing report (COBOL only) and record layouts, for listing Data and Procedure cross references, and for creating both a detailed flowchart and a logic chart. AUTODOC can process programs individually, or it can process up to 99 source programs in the batch mode.

AUTODOC can operate on the following systems: Honeywell 200 central processor with 28K characters of core, IBM System 360 (model 25 and up) configured with a 48K problem program partition and capable of operating in a DOS or OS environment, CDC Series 3304 or 3504 central processor with 32K words of core, and NCR century 100 or 200 with 32K core, and a Burroughs B 5500 with 64 K core. AUTODOC is written in COBOL.

Cost for AUTODOC is $4,800 for a 3-year license agreement for the first customer installation. Each additional installation of the same language costs $2,500. Separate AUTODOC packages for processing COBOL and assembly language can be purchased and the package for processing the second language costs $1,800.

PACKAGE OUTPUT

The cover page gives the name and author of the program. It also gives information describing the program's hardware environment, security status, and origination date. The remarks section can give a complete narrative description of the program.

The source program list lists the statements processed along with their sequence number. Thus number can be used as a reference number late by autodoc.

The error list gives certain syntax errors found. It lists the statement in error and the sequence number of this statement.

The report layout (COBOL only) is provided for all reports defined in the Report Section.

The data reference list (COBOL only) lists all data items used in the program along with information pertaining to each of them.

The procedure reference list (COBOL only) lists alphabetically all procedure names used in the program along with its associated source sequence number. Also given are the page connector number of the flow-chart connector symbol generated by the procedure name and the source sequence numbers of statements which reference it.

The special reference list (COBOL only) lists all source sequence numbers referencing an external name, a literal, figurative constants, and system names.

The label reference list (assembly only) gives an alphabetical listing of all labels defined within the CSECT or DSECT bring processed. Their associated sequence number is also given. Additional entries include the page and connector symbol generated by the label, and the

source sequence numbers of all statements that reference that label name.

The detail flowchart provides a two-dimensional representation of the logic flow. It constructs a symbol for each source statement and its related text. The user can select ANSI Standard or IBM flowchart symbols. Each flowchart logical page is divided vertically into four position segments, with flow direction from top to bottom. The flowchart can be printed as one or two physical pages per logical page. Both offpage and onpage connectors are generated.

The logic chart (COBOL only) charts only statements which affect the logical flow of the program, as well as statements of the notes and input/output type.

AUTODOC

D O C U M E N T A T I O N   O F

'AUTODOC SAMPLE'

A U T H O R — CTC COMPUTER CORPORATION, PROPRIETARY SYSTEMS DIV.

I N S T A L L A T I O N — CTC - PSD.

S E C U R I T Y — THIS IS A PROPRIETARY PROGRAM, THE USE OF WHICH IS GOVERNED BY CONTRACTUAL AGREEMENT.

D A T E   W R I T T E N — 1969.

R E M A R K S — AUTODOC IS A TOTAL PROGRAM DOCUMENTATION SYSTEM. IT PROVIDES AUTOMATICALLY THE FOLLOWING PRODUCT OUTPUTS -

1) COVER PAGE
2) RECORD LAYOUT
3) SOURCE LISTING
4) ERROR LIST
5) DATA REFERENCE LIST
6) PROCEDURE REFERENCE LIST
7) SPECIAL REFERENCE LIST
8) DETAILED FLOWCHART
9) LOGIC CHART

AUTODOC IS AN EXTREMELY EASY SYSTEM TO USE - NO SPECIAL CODING OR INPUT PREPARATION IS REQUIRED. ALL PRODUCTS ARE AUTOMATICALLY GENERATED UNLESS SPECIFICALLY SUPPRESSED THROUGH THE USE OF SIMPLE PARAMETER CARD(S).

P R O P R I E T A R Y   S Y S T E M S   D I V I S I O N

```
PROGRAM 'AUTODOC SAMPLE'          ** S O U R C E   P R O G R A M   L I S T **                    PAGE   1
------------------------------------------------------------------------------------------------------
REF NBR:CPY:                                  SOURCE STATEMENT
------------------------------------------------------------------------------------------------------
       :   :
     1 :   : 000000 IDENTIFICATION DIVISION.                                                   SAMPLE
     2 :   : 000010 PROGRAM-ID. 'AUTODOC SAMPLE'.                                              SAMPLE
     3 :   : 000020 AUTHOR.  CTC COMPUTER CORPORATION, PROPRIETARY SYSTEMS DIV.                SAMPLE
     4 :   : 000030 INSTALLATION.  CTC - PSD.                                                  SAMPLE
     5 :   : 000040 DATE-WRITTEN.  1969.                                                       SAMPLE
     6 :   : 000050 SECURITY.   THIS IS A PROPRIETARY PROGRAM, THE USE OF WHICH                SAMPLE
     7 :   : 000060             IS GOVERNED BY CONTRACTUAL AGREEMENT.                          SAMPLE
     8 :   : 000070 REMARKS.     AUTODOC IS A TOTAL PROGRAM DOCUMENTATION SYSTEM.              SAMPLE
     9 :   : 000080              IT PROVIDES AUTOMATICALLY THE FOLLOWING PRODUCT               SAMPLE
    10 :   : 000090                  OUTPUTS -                                                 SAMPLE
    11 :   : 000100                                                                            SAMPLE
    12 :   : 000110                          1)  COVER PAGE                                    SAMPLE
    13 :   : 000120                          2)  RECORD LAYOUT                                 SAMPLE
    14 :   : 000130                          3)  SOURCE LISTING                                SAMPLE
    15 :   : 000140                          4)  ERROR LIST                                    SAMPLE
    16 :   : 000150                          5)  DATA REFERENCE LIST                           SAMPLE
    17 :   : C00160                          6)  PROCEDURE REFERENCE LIST                      SAMPLE
    18 :   : 000170                          7)  SPECIAL REFERENCE LIST                        SAMPLE
    19 :   : 000180                          8)  DETAILED FLOWCHART                            SAMPLE
    20 :   : 000190                          9)  LOGIC CHART                                   SAMPLE
    21 :   : 000200                                                                            SAMPLE
    22 :   : 000210              AUTODOC IS AN EXTREMELY EASY SYSTEM TO USE -                  SAMPLE
    23 :   : 000220                  NO SPECIAL CODING OR INPUT PREPARATION IS                 SAMPLE
    24 :   : 0C0230                  REQUIRED.  ALL PRODUCTS ARE AUTOMATICALLY                 SAMPLE
    25 :   : 000240                  GENERATED UNLESS SPECIFICALLY SUPPRESSED                  SAMPLE
    26 :   : 000250                  THROUGH THE USE OF SIMPLE PARAMETER CARD(S).              SAMPLE
    27 :   : 000260 ENVIRONMENT DIVISION.                                                      SAMPLE
    28 :   : 000270 SPECIAL-NAMES. 'SYSIN' IS CRD-RDR.                                         SAMPLE
    29 :   : 000280 CONFIGURATION SECTION.                                                     SAMPLE
    30 :   : 000290 SOURCE-COMPUTER.    IBM-360.                                               SAMPLE
    31 :   : 000300 INPUT-OUTPUT SECTION.                                                      SAMPLE
    32 :   : 000310 FILE-CONTROL.                                                              SAMPLE
    33 :   : 000320     SELECT MAST-IN ASSIGN TO 'SYS050' UNIT-RECORD 2540R UNIT.              SAMPLE
    34 :   : 000330 DATA DIVISION.                                                             SAMPLE
    35 :   : 000340 FILE SECTION.                                                              SAMPLE
    36 :   : 000350 FD  MAST-IN                                                                SAMPLE
    37 :   : 000360     RECORDING MODE IS F                                                    SAMPLE
    38 :   : 000370     LABEL RECORDS ARE OMITTED                                              SAMPLE
    39 :   : 000380     RECORD CONTAINS 80 CHARACTERS                                          SAMPLE
    40 :   : 000390     DATA RECORDS ARE MAST-IN, MAST-OUT.                                    SAMPLE
    41 :   : 000665 01  COPY 'TEST'.                                                           SAMPLE
    42 : * :            02  C PICTURE X.
    43 : * :            02  D PICTURE X.
    44 : * :            02  E PICTURE X(10).
    45 : * :            02  F PICTURE X(10).
    46 :   : 000400 01  MAST-IN.                                                               SAMPLE
    47 :   : 000410     02  A PICTURE X.                                                       SAMPLE
    48 :   : 000420     02  B PICTURE X.                                                       SAMPLE
    49 :   : 000430 01  MAST-OUT.                                                              SAMPLE
    50 :   : 000440     02  A PICTURE X.                                                       SAMPLE
    51 :   : 000450     02  B PICTURE X.                                                       SAMPLE
    52 :   : 000460     02  C PICTURE X.                                                       SAMPLE
    53 :   : 0C0470     02  D PICTURE X.                                                       SAMPLE
    54 :   : 000480             88  YES VALUE 0.                                               SAMPLE
    55 :   : 000490             88  NO  VALUE 1.                                               SAMPLE
    56 :   : 000500 WORKING-STORAGE SECTION.                                                   SAMPLE
    57 :   : 000510 77  WS-CNT-1 PICTURE 9.                                                    SAMPLE
    58 :   : 000520 77  WS-CNT-2 PICTURE 9.                                                    SAMPLE
    59 :   : 000530 77  WS-CNT-3 PICTURE 99.                                                   SAMPLE
    60 :   : 000540 77  S PICTURE 9.                                                           SAMPLE
    61 :   : 000550 77  RCDS-IN     VALUE ZERO  PICTURE S9(3), COMPUTATIONAL-3.                SAMPLE
    62 :   : 000560 77  RCDS-OUT    VALUE ZERO  PICTURE S9(3), COMPUTATIONAL-3.                SAMPLE
    63 :   : 000570 77  QTY-1                     PICTURE ****,***.99-.                        SAMPLE
    64 :   : 000580 77  QTY-2                     PICTURE ****,***.99-.                        SAMPLE
    65 :   : 000590 01  TEMP-REC.                                                              SAMPLE
    66 :   : 0C0600     02  A PICTURE X OCCURS 5.                                              SAMPLE
    67 :   : 000610     02  B PICTURE X OCCURS 5.                                              SAMPLE
       :   :
------------------------------------------------------------------------------------------------------
```

```
PPOGRAM  'AUTODOC SAMPLE'                    ** E R R O R   L I S T **                                              PAGE    1

------------------------------------------------------------------------------------------------------------------------------
REF NBR:           ERROR DESCRIPTION         :                                   SOURCE IMAGE
------------------------------------------------------------------------------------------------------------------------------
      84:  SYNTAX ERROR...SISPLAY            :                   'OH HAPPY DAY' UPON CONSOLE.
------------------------------------------------------------------------------------------------------------------------------
```

B-34

PROGRAM 'AUTODOC SAMPLE' ** R E C O R D   L A Y O U T **

FILE NAME	REF NBR	MODE	RECORD SIZE	BLOCKED	FILE SIZE
MAST-IN	36	F	80	1	

BYTE	RECORD/DATA NAME	REF NBR	TYPE	FORM	LOG SIZE	PHYS SIZE	OCCURS	JUST	SYNC	SGN	POINT LOC	ZERO SUPP	CHK	FLT	LV	BLNK
	COPY	41	RCD													
1	C	42	ELEM	A/N DISP	1	1										
2	D	43	ELEM	A/N DISP	1	1										
3	E	44	ELEM	A/N DISP	10	10										
13	F	45	ELEM	A/N DISP	10	10										
	MAST-IN	46	RCD													
1	A	47	ELEM	A/N DISP	1	1										
2	B	49	ELEM	A/N DISP	1	1										
	MAST-OUT	49	RCD													
1	A	50	ELEM	A/N DISP	1	1										
2	B	51	ELEM	A/N DISP	1	1										
3	C	52	ELEM	A/N DISP	1	1										
4	D	53	ELEM	A/N DISP	1	1										

WORKING-STORAGE

BYTE	RECORD/DATA NAME	REF NBR	TYPE	FORM	LOG SIZE	PHYS SIZE	OCCURS	JUST	SYNC	SGN	POINT LOC	ZERO SUPP	CHK	FLT	LV	BLNK
	TEMP-REC	65	RCD													
1	A	66	ELEM	A/N DISP	1	1	5									
6	B	67	ELEM	A/N DISP	1	1	5									
11	K	68	ELEM	A/N DISP	1	1	5									
16	L	69	ELEM	A/N DISP	1	1	5									
	THE-WORLD	70	RCD													
1	THIS-IS-THE-LONGEST-NAME	71	ELEM	A/N DISP	1	1										
	H-STORAGE	72	RCD													
1	H1	73	ELEM	A/N DISP	1	1										
2	H2	74	ELEM	A/N DISP	1	1										
3	H3	75	ELEM	A/N DISP	1	1										
4	AN-PRICE	76	ELEM	A/N DISP	10	10										
4	PRICE	77	ELEM	NUM DISP	10	10				X	L 4					
14	PRICES	79	ELEM	NUM DISP	14	14					L 5	X		$		
28	DUMMY-DATA-NAME	80	RCD	A/N DISP	1	1										

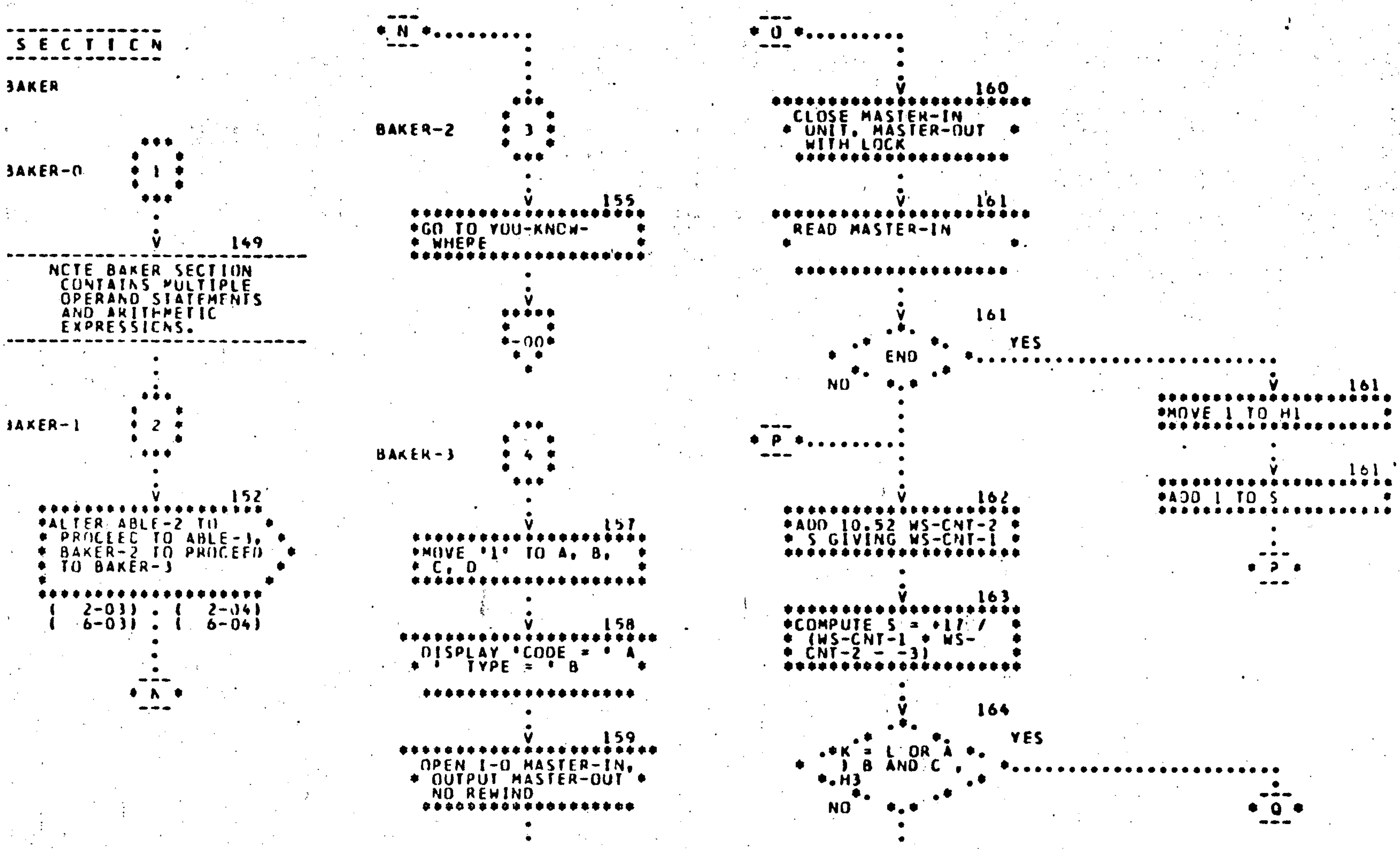
PROGRAM "AUTODOC SAMPLE" **D A T A R E F E R E N C E L I S T** PAGE 1

NAME	REF NBR	TYPE	REFERENCES
	UNDEFINED		212
A	47	02	NO REF
A	50	02	NO REF
A	66	02	127 155 156 162 199 200
AN-PRICE	76	02	77
B	48	02	NO REF
B	51	02	NO REF
B	67	02	127 155 156 162 199
C	42	02	NO REF
C	52	02	133 155 162 201 202
COPY	41	01	NO REF
CRD-RDR	28	SPL	NO REF
D	43	02	NO REF
D	53	02	133 155
DUMMY-DATA-NAME	80	01	NO REF
E	44	02	128
F	45	02	129 130
H-STORAGE	72	01	NO REF
H1	73	02	159 199 200 203 213
H2	74	02	135 199 201 202 203 206
H3	75	02	135 162 201 202 204 206
K	68	02	133 162 203 204
L	69	02	162 204
MAST-IN	36	FD	NO REF
MAST-IN	46	01	33 40 103 112 114 118 121
MAST-OUT	49	01	40 103 110 111 112 113 114 118 121 186
MASTER-IN	UNDEFINED		105 107 109 110 157 158 159 186
MASTER-OUT	UNDEFINED		106 108 157 158
NO	55	88	NO REF
PRICE	77	02	NO REF
PRICES	79	02	NO REF
QTY-1	63	77	NO REF
QTY-2	64	77	NO REF
RCDS-IN	61	77	NO REF

NAME	REF NBR	CMT LOC	REFERENCES					
ABLE	85	2-01	NO REF					
ABLE-0	86	2-01	NO REF					
ABLE-1	89	2-02	132	136	137	138	139	141
ABLE-2	91	2-03	90	135	150			
ABLE-3	94	2-04	92	150				
ABLE-4	97	2-05	90	95				
ABLE-5	98	2-05	95					
ABLE-6	115	2-06	95					
BAKER	145	4-01	NO REF					
BAKER-0	146	4-01	NO REF					
BAKER-1	149	4-02	NO REF					
BAKER-2	152	4-03	150					
BAKER-3	154	4-04	151					
BAKER-4	164	5-01	162					
CHARLEY	171	6-01	NO REF					
CHARLEY-0	172	6-01	NO REF					
CHARLEY-1	175	6-02	163	167	194			
CHARLEY-2	178	6-03	166					
CHARLEY-3	180	6-04	167					
CHARLEY-4	182	6-05	NO REF					
DOG	195	7-01	NO REF					
DOG-0	196	7-01	NO REF					
DOG-1	198	7-02	NO REF					
EASY	205	8-01	NO REF					
EASY-0	206	8-01	NO REF					
EASY-1	209	8-02	NO REF					
EASY-3	219	8-03	NO REF					
MIRACLES	UNDEFINED		83					
START	82	1-01	NO REF					
ABLE-3 IN ABLE	94	2-04	215					
CHARLEY-2 IN CHARLEY	178	6-03	176	186				
CHARLEY-3 IN CHARLEY	180	6-04	176	179				

EXTERNAL NAMES		REFERENCES									
'FOR PHILIP MORRIS'		100									

LITERALS, FIG CONS, SYSTEM NAMES		REFERENCES									
.0079		166									
.15		125									
.3		165									
+17		161									
-1		168									
-3		161									
' TYPE = '		156									
'.'		184									
'@'		130									
'CODE = '		156									
'MERRY CHRISTMAS'		104									
'THIS IS TO TEST FOR THE FLOAT		217									
'X'		129									
'Y'		129									
'1'		155									
QUOTE		130									
SPACES		128									
TALLY		144									
ZERO		138	168	202							
ZEROS		128									
0		112									
1		117	120	134	139	139	159	159	201	203	211
10		141	167								
10.52		160									
100		123									



SECTION
BAKER
BAKER-0
1
149
NOTE BAKER SECTION CONTAINS MULTIPLE OPERAND STATEMENTS AND ARITHMETIC EXPRESSIONS.
BAKER-1
2
152
ALTER ABLE-2 TO PROCEED TO ABLE-1, BAKER-2 TO PROCEED TO BAKER-3
(2-03) . (2-04)
(6-03) . (6-04)
N
N
BAKER-2
3
155
GO TO YOU-KNOW-WHERE
00
BAKER-3
4
157
MOVE '1' TO A, B, C, D
158
DISPLAY 'CODE = ' A ' TYPE = ' B
159
OPEN I-O MASTER-IN, OUTPUT MASTER-OUT NO REWIND
O
160
CLOSE MASTER-IN UNIT, MASTER-OUT WITH LOCK
161
READ MASTER-IN
161
END
YES
NO
161
MOVE 1 TO H1
161
ADD 1 TO S
P
P
162
ADD 10.52 WS-CNT-2 S GIVING WS-CNT-1
163
COMPUTE S = +17 / (WS-CNT-1 + WS-CNT-2 - -3)
164
K = L OR A) B AND C .H3
YES
NO
Q

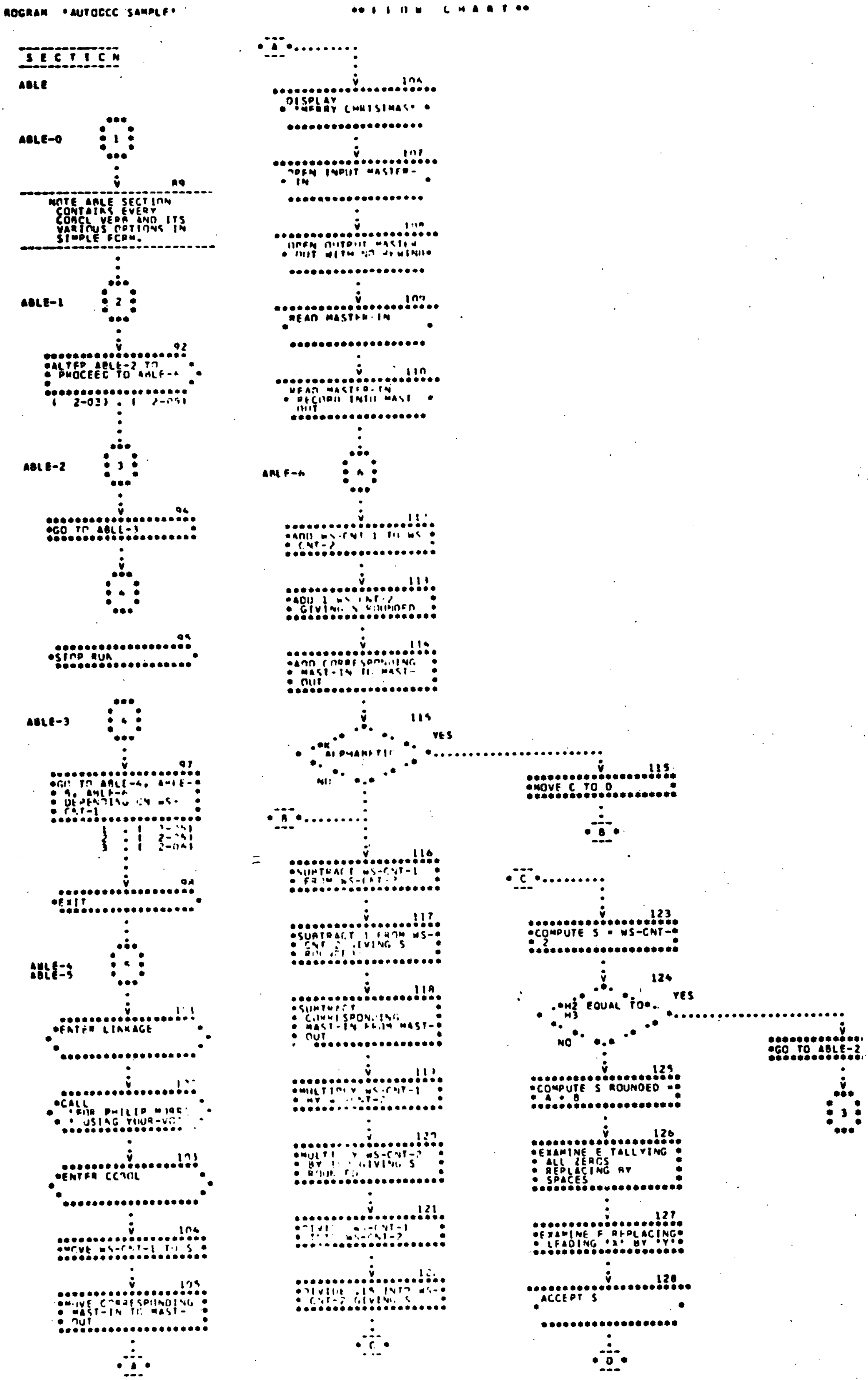
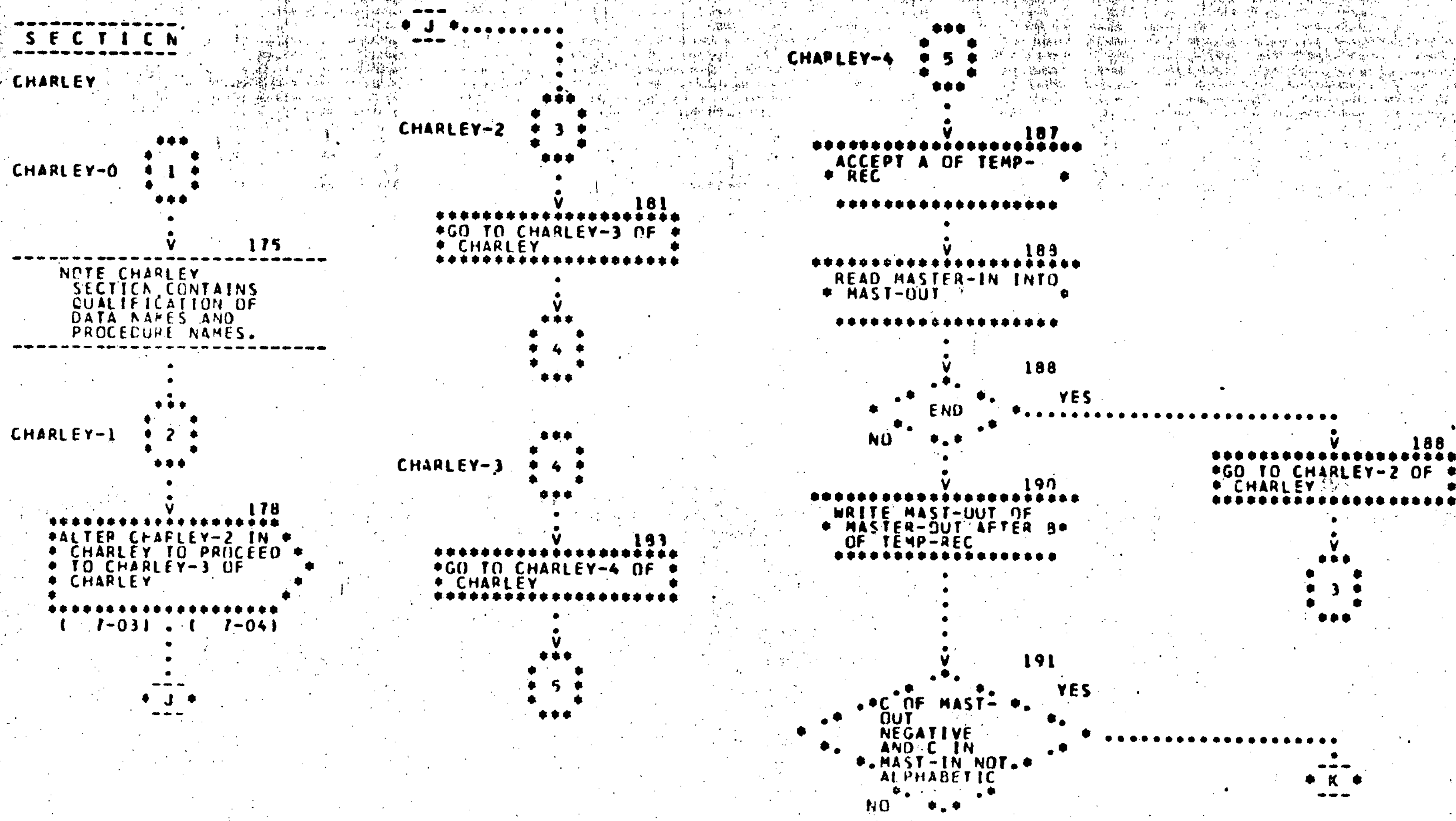

ROGRAM 'AUTOCCC SAMPLE'
** F L O W C H A R T **
SECTION
ABLE
ABLE-0
NOTE ABLE SECTION CONTAINS EVERY COBOL VERB AND ITS VARIOUS OPTIONS IN SIMPLE FORM.
ABLE-1
ALTER ABLE-2 TO PROCEED TO ABLE-4
ABLE-2
GO TO ABLE-3
STOP RUN
ABLE-3
EXIT
ENTER LINKAGE
ENTER COBOL
DISPLAY 'MERRY CHRISTMAS'
OPEN INPUT MASTER-IN
READ MASTER-IN
ADD CORRESPONDING MAST-IN TO MAST-OUT
ALPHABETIC
YES
NO
MOVE C TO D
COMPUTE S = WS-CNT-2
H2 EQUAL TO H3
GO TO ABLE-2
COMPUTE S ROUNDED = A + B
EXAMINE E TALLYING ALL ZEROS REPLACING BY SPACES
EXAMINE F REPLACING LEADING 'X' BY 'Y'
ACCEPT S

FIGURE 11


S E C T I C N
CHARLEY
CHARLEY-0
1
175
NOTE CHARLEY SECTICN CONTAINS QUALIFICATION OF DATA NAMES AND PROCEDURE NAMES.
CHARLEY-1
2
178
ALTER CHARLEY-2 IN CHARLEY TO PROCEED TO CHARLEY-3 OF CHARLEY
(7-03) . (7-04)
J
J
CHARLEY-2
3
181
GO TO CHARLEY-3 OF CHARLEY
4
CHARLEY-3
4
183
GO TO CHARLEY-4 OF CHARLEY
5
CHARLEY-4
5
187
ACCEPT A OF TEMP-REC
188
READ MASTER-IN INTO MAST-OUT
188
END
YES
NO
188
GO TO CHARLEY-2 OF CHARLEY
3
190
WRITE MAST-OUT OF MASTER-OUT AFTER B OF TEMP-REC
191
C OF MAST-OUT NEGATIVE AND C IN MAST-IN NOT ALPHABETIC
YES
NO
K

AUTOFLOW

APPLIED DATA RESEARCH

GENERAL

AUTOFLOW translates source language programs, written in COBOL, FORTRAN, PL/1, assembly language or numerous autocode languages, into flowchart documents in various levels of detail. This includes statement analysis, page allocation, line drawing, and rearrangement of source input as necessary.

The package runs on IBM 360 Series (under OS, DOS, TOS), IBM 1400 Series, IBM 7090 Series, RCA Spectra 70 Series (under TDOS), Honeywell 200 Series; requires one tape or disc and printer, microfilm, or plotter. AUTOFLOW is a single, multiphase program written in BAL.

A permanent license costs $3,000 to $7,000 depending on the language features required.

An unlimited monthly usage license is priced at a flat monthly rate depending on the language features required.

PACKAGE OUTPUT

The AUTOFLOW chart set is produced which includes:

- Title Sheet - This listing contains the program name, date, and other pertinent information.
- Input Listing - This printout contains a complete 80/80 listing of the input program.
- Procedural Statement Label Index - This listing specifies appropriate section and paragraph names, labels, or statement numbers in alphabetic order and provides a quick reference between the source program and the flowchart.

Table of Contents and References - This cross-reference table provides indexing information for locating transfers of control, both within the flowchart and the source program, whether the references are explicit or implicit.

Table of Diagnostics - This listing contains a record of logical-flow errors, incomplete paths, missing references, and other programming errors.

Flowchart - Each flowchart covers two consecutive printer sheets and can contain up to four columns of flow paths. The symbols on each page are numbered consecutively.

SPECIAL LISTINGS FOR COBOL:

COBOL Diagnostic Analysis - Analysis of the COBOL program is performed identifying logical flow errors, as well as syntax errors.

Procedure Division Analysis - Provides a summary of the various vital activities which take place in the Procedure Division.

Data Division Analysis - (Data Cross Reference) shows the flowchart locations and source sequence numbers for each data name reference in the program.

Data Record Map - Presents a descriptive layout of all group and elementary items in the records within all sections of the Data Division.

Data Division Index - Contains all data and mnemonic name items sequenced alphanumerically, as well as pertinent information for each item.

High Level Flowchart - The COMPRESS facility of Autoflow system enables a user to control the level of flowchart detail.

SPECIAL LISTINGS FOR ASSEMBLY:

EQU Statements - A chronological collection of special symbols to represent all EQU statements used in the source program.

Constants and Storage Areas Listings - A listing of all constants and storage areas.

Modified Tag Summary - A listing of all modified tag references, as well as their location.

Literal Summary - A listing of all literals used in the source program as well as the sequence number in which each literal appears.

Macro Usage Summary - A listing of all macros used in the program, of where they were invoked, and of where the applicable definition is located.

SPECIAL LISTINGS FOR PL/I:

On-Unit Action Blocks - Statements which comprises interrupt condition specifications are flowcharted as separate units.

Called Procedures Cross Reference - This chart provides a summary display of all CALLED entry points in the source input.

Signalled On-Unit Action Blocks - This chart graphically represents all signalled interrupt conditions and their points of reference.

Label Assignment Cross Reference - This is a chart to illustrate the effect of label variables used in GO TO statements on the logical flow of a program.

Duplicate Declaration Map - Multiple declaration of an identifier are listed.

Condition Prefix Map - This illustrates the physical placement of condition prefixes within the nested procedures and BEGIN blocks in the source input.

Declaration Statements - This listing displays non-procedural declaration statements in the order of their appearance in the source input.

GET/PUT, FORMAT Statements -These type statements are printed in this listing.

Note: Each language has the CHART option which enables the user to control the level of detail in the flowchart.

Comment: Autoflow seems to be the most complete and successful proprietary automative documentation package on the market today. It by far has the most installations of any of the documentation packages, and has more features than any other package.

COBOL MODULE (LIST1,NAMSQ,PARM)

CARD NO **** CONTENTS ****

```
 OPTION DXREF=YES,PMAP=YES,DMAP=YES,CINDEX=YES
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. ADR-SAMPLE.
000300 AUTHOR. WICK.
000400 INSTALLATION. PRINCETON RESEARCH CENTER.
000500 DATE-WRITTEN. APRIL 1, 1967.
000600 DATE-COMPILED. TODAY.
000601 REMARKS. AN AUTOFLOW CHART PRODUCED FROM COBOL SOURCE INPUT
000602     MAY CONTAIN UP TO FIVE PARTS. THE LISTING OF THE SOURCE
000603     INPUT, THE TABLE OF CONTENTS AND CROSS REFERENCE LISTING,
000604     THE TABLE OF DIAGNOSTICS, A 'REMARKS' PORTION AND THE FLOW
000605     CHART PRODUCED FROM THE PROCEDURE DIVISION OF THE SOURCE
000606     PROGRAM.
000607
000608     THE LISTING OF THE SOURCE INPUT IS OPTIONAL, SELECTED BY
000609     THE USER IN HIS PARAMETER CARD. THE INPUT LIST OPTION
000610     HAS NOT BEEN INVOKED FOR THE AUTOFLOW CHARTING OF THIS
000610     PROGRAM.
000611
000612     THE TABLE OF CONTENTS INDICATES THE CHART LOCATION OF
000613     EACH SECTION OR PARAGRAPH WITHIN THE PROGRAM AS WELL AS
000614     ALL REFERENCES TO THE SECTION OR PARAGRAPH PRODUCED FROM
000615     THE GO TO, PERFORM, ALTER, AND PROCESS VERBS. IT ALSO
000616     INDICATES THE SOURCE CARD NUMBER (IN THE NAME FIELD) OF
000617     ALTER STATEMENTS ALONG WITH THE CHART LOCATION OF THE
000618     STATEMENT BEING ALTERED. CROSS REFERENCES PRODUCED BY
000619     LOCAL DECISIONS, THOSE DECISIONS WHOSE PATHS COME
000620     TOGETHER AT 'NEXT SENTENCE', ARE INDICATED WITH A BLANK
000621     NAME FIELD.
000622
000622     WHEN PRESENT, THE 'REMARKS' STATEMENT IN THE
000623     IDENTIFICATION DIVISION WILL APPEAR AS INTRODUCTORY
000624     NARRATIVE, IN THE TEXT FORMAT OF AN AUTOFLOW FLOW CHART.
000625     IF NO 'REMARKS' STATEMENT IS PRESENT, THIS NARRATIVE
000626     CHART IS NOT PRESENT. THE MATERIAL CURRENTLY BEING
000627     PRINTED IS AN EXAMPLE OF THE 'REMARKS' STATEMENT NARRATIVE
000529     CHART
000628
000629     EACH PROCEDURE DIVISION 'SECTION' PRODUCES A CHART, WHOSE
000630     CHART TITLE IS THE SECTION NAME. THIS CHART TITLE APPEARS
000631     IN THE HEADING OF EACH FLOW CHART PAGE PRODUCED FOR THIS
000632     SECTION'S PROCEDURES. AS EACH SECTION BECOMES A CHART,
000633     THE FLOW CHARTS FOR ANY SECTION MAY BE LOCATED FROM THE
000634     TABLE OF CONTENTS.
000635
000636     WHEN A PROGRAM IMPLIES A BRANCH FROM ONE SECTION TO THE
000637     NEXT, A BRANCH SYMBOL IS GENERATED IN THE FLOW CHART.
000638
000639
000700     THIS PROGRAM IS INTENDED ONLY AS A SAMPLE OF A PROGRAM
000800     WHICH MIGHT BE RUN UNDER AUTOFLOW. IT PROBABLY IS NOT
000900     SUBJECT TO CLOSE SCRUTINY FOR LOGICAL PROGRAM CAPABILITIES.
001000
001100 ENVIRONMENT DIVISION.
001200 CONFIGURATION SECTION.
001300 SOURCE-COMPUTER. IBM-360 F30.
001400 OBJECT-COMPUTER. IBM-360 F30.
001500
001600 INPUT-OUTPUT SECTION.
001700 FILE-CONTROL.
001800     SELECT TRANS-IN ASSIGN TO 'SYS011'.
001900     SELECT MASTER-IN ASSIGN TO 'SYS012'.
002000     SELECT MASTER-OUT ASSIGN TO 'SYS013'.
002100
002200 DATA DIVISION.
002300 FILE SECTION.
002400 FD  MASTER-IN
002500     BLOCK CONTAINS 2000 CHARACTERS
002600     RECORD CONTAINS 100 CHARACTERS
002700     LABEL RECORDS ARE STANDARD
002800     DATA RECORDS ARE MASTIN.
002900 01 MASTIN.
003000     02 MASTIN-KEY.
003100        03 INACCT   PICTURE IS X(10).
003200        03 INTYPE   PICTURE IS X(2).
003300        03 INREF    PICTURE IS X(16).
003400        03 INCODE   PICTURE IS X.
003500     02 MASTIN-NAME-ADDRESS.
003600        03 INNAME   PICTURE IS X(26).
003700        03 INADDRESS  PICTURE IS X(23).
003800        03 INCITY   PICTURE IS X(15).
003900        03 INSTATE   PICTURE IS X(2).
004000        03 INZIP    PICTURE IS X(5).
004100 FD MASTER-OUT
004200     BLOCK CONTAINS 2000 CHARACTERS
004300     RECORD CONTAINS 100 CHARACTERS
004400     LABEL RECORDS ARE STANDARD
004500     DATA RECORDS ARE MASTOUT.
004600 01 MASTOUT.
004700     02 MASTOUT-KEY.
004800        03 OUTACCT PICTURE IS X(10).
004900        03 OUTYPE  PICTURE IS X(2).
005000        03 OUTREF  PICTURE IS X(16).
005100        03 OUTCODE PICTURE IS X.
005200     02 MASTOUT-NAME-ADDRESS.
005300        03 OUTNAME PICTURE IS X(26).
005500        03 OUTADDRESS PICTURE IS X(23).
005600        03 OUTCITY PICTURE IS X(15).
005700        03 OUTSTATE PICTURE IS X(2).
005800        03 OUTZIP  PICTURE IS X(5).
005900 FD TRANS-IN
006000     BLOCK CONTAINS 2000 CHARACTERS
006100     RECORD CONTAINS 100 CHARACTERS
006200     DATA RECORDS ARE TRANSIN
006300     LABEL RECORDS ARE STANDARD.
006400 01 TRANSIN.
006500     02 TRANSIN-KEY.
006600        03 TRACCT   PICTURE IS X(10).
006700        03 TRTYPE   PICTURE IS X(2).
006800        03 TREF     PICTURE IS X(16).
006900        03 TRCODE   PICTURE IS X.
007000     02 TRANS-NAME-ADDRESS.
007100        03 TRNAME   PICTURE IS X(26).
007200        03 TRADDRESS  PICTURE IS X(23).
007300        03 TRCITY   PICTURE IS X(15).
007400        03 TRSTATE   PICTURE IS X(2).
007500        03 TRZIP    PICTURE IS X(5).
007600
007700 WORKING-STORAGE SECTION.
007800 77 PREV-MASTIN        PICTURE IS X(36).
007900 77 PREV-MASTOUT       PICTURE IS X(36).
008000 77 PREV-TRANSIN       PICTURE IS X(36).
008100 77 EOF-SW    PICTURE IS X VALUE IS SPACE.
008200 77 OUTAREA-SW PICTURE IS X VALUE IS SPACE.
008300 77 ERROR-TYPE PICTURE IS X(40).
008400 01 CURR-MASTIN.
008500     02 CURR-MASTIN-KEY.
008600        03 CURR-MASTIN-ACCT    PICTURE IS X(10) VALUE IS SPACES.
008700        03 CURR-MASTIN-TYPE    PICTURE IS X(2)  VALUE IS SPACES.
008800     02 CMAK REDEFINES CURR-MASTIN-KEY PICTURE IS X(12).
008900     02 CURR-MASTIN-REFER      PICTURE IS X(16) VALUE IS SPACES.
009000     02 CURR-MASTIN-CODE       PICTURE IS X     VALUE IS SPACE.
009100 01 CURR-MASTOUT.
009200     02 CURR-MASTOUT-KEY.
009300        03 CURR-MASTOUT-ACCT  PICTURE IS X(10) VALUE IS SPACES.
009400        03 CMA REDEFINES CURR-MASTOUT-ACCT PICTURE IS X(10).
009500        03 CURR-MASTOUT-TYPE  PICTURE IS X(2)  VALUE IS SPACES.
009600     02 CMK REDEFINES CURR-MASTOUT-KEY PICTURE IS X(12).
009700     02 CURR-MASTOUT-REFER    PICTURE IS X(16) VALUE IS SPACES.
009800     02 CURR-MASTOUT-CODE     PICTURE IS X     VALUE IS SPACE.
009900     02 CMC REDEFINES CURR-MASTOUT-CODE PICTURE IS X.
010000 01 PREV-MASTOUT.
010100     02 PREV-MASTOUT-KEY.
010200        03 PREV-MASTOUT-ACCT  PICTURE IS X(10) VALUE IS SPACES.
010300        03 PREV-MASTOUT-TYPE  PICTURE IS X(2)  VALUE IS SPACES.
010400     02 PREV-MASTOUT-REFER    PICTURE IS X(16) VALUE IS SPACES.
010500     02 PREV-MASTOUT-CODE     PICTURE IS X     VALUE IS SPACE.
010600 01 TRAN-SORT-KEY.
010700     02 TRAN-SEQ-KEY.
010800        03 TRAN-ACCT.
010900           04 TRAN-ACCT-KEY.
011100              05 TACCT-NO        PICTURE IS X(10).
011200              05 TAN REDEFINES TACCT-NO PICTURE IS X(10).
```

CARD NO **** CONTENTS ****

```
011300          05 TACCT-TYPE         PICTURE IS X(2).
011350       0- TAK REDEFINES TRAN-ACCT-KEY PICTURE IS X(12).
011400          03 TRIFFERENCE        PICTURE IS X(16).
011500          03 TACCT-CODE         PICTURE IS X.
011600          03 TAC REDEFINES TACCT-CODE PICTURE IS X.
011700       02 TSK REDEFINES TRAN-SEQ-KEY PICTURE IS X(29).
011800       02 TRANS-CODE.
011900          03 TC1                PICTURE IS 9.
012000          03 TC2                PICTURE IS 9.
012200       02 TAMT                  PICTURE IS S9(5)V99.
012300
012400 PROCEDURE DIVISION.
012500 START.
012600     OPEN INPUT MASTER-IN.
012700     OPEN INPUT TRANS-IN.
012800     OPEN OUTPUT MASTER-OUT.
012900     PERFORM GET-MASTER-ROUTINE THROUGH GET-MASTER-EXIT.
013000 READ-NEXT-TRANS.
013100     READ TRANS-IN,
013200         AT END  GO TO TRANS-EOF.
013300     MOVE T-KEY TO TRAN-SORT-KEY.
013400     IF TRAN-SORT-KEY IS NOT LESS THAN PREV-TRANSIN
013500     GO TO MOVE-KEY, ELSE
013600         DISPLAY 'TRANSACTION-FILE OUT-OF-SEQ' UPON CONSOLE,
013700         STOP RUN.
013800 TRANS-EOF.
013900     CLOSE TRANS-IN WITH LOCK.
014000     IF EOF-SW IS NOT EQUAL TO SPACES
014100     GO TO END-OF-JOB,
014200     ELSE MOVE 'T' TO EOF-SW.
014300     MOVE HIGH-VALUE TO TRAN-SORT-KEY.
014400 MOVE-KEY.
014500     MOVE TRAN-SORT-KEY TO PREV-TRANSIN.
014600 TEST-OUTAREA.
014700     IF   OUTAREA-SW IS EQUAL TO SPACES
014800          GO TO COMPARE-INPUTS.
014900     IF   CURR-MASTOUT IS EQUAL TO TRAN-SEQ-KEY
015000          GO TO TEST-TRANS-CODE.
015100
015200     NOTE  **TRANSACTION IS HIGH TO OUTPUT MASTER RECORD.
015300
015400     PERFORM PUT-MASTER THROUGH PUT-MASTER-EXIT UNTIL OUTACCT I
015500     S NOT LESS THAN TRACCT.
015600
015700 COMPARE-INPUTS.
015800     IF TRAN-SEQ-KEY IS LESS THAN CURR-MASTIN GO TO TRANS-LOW,
015900     ELSE PERFORM PUT-MASTER THROUGH PUT-MASTER-EXIT.
016000     IF TSK IS EQUAL TO CMK GO TO TEST-TRANS-CODE.
016100     GO TO TEST-OUTAREA.
016200 TRANS-LOW.
016300
016400     NOTE  **TRANSACTION IS LOW TO INPUT MASTER RECORD.
016500
016600 TRAANS-LOW.
016700     IF TAN IS NOT EQUAL TO CMA
016800         ALTER END-OF-ACCT-EXIT TO PROCEED TO TEST-TRANS-CODE.
016900         GO TO END-OF-ACCT-ROUTINE,
017000     ELSE IF TAC IS NOT EQUAL TO CMC
017100     GO TO TEST-TRANS-CODE ELSE
017200         ALTER END-OF-CODE-EXIT TO PROCEED TO TEST-TRANS-CODE.
017300 END-OF-CODE-ROUTINE.
017400     IF OUTAREA-SW IS NOT EQUAL TO SPACES, WRITE MASTER-OUT,
017500     MOVE SPACES TO OUTAREA-SW, MASTOUT-NAME-ADDRESS.
017600     IF TC1 IS EQUAL TO 1 MOVE 'ERROR DELETE RECORD' TO
017700     ERROR-TYPE AND ALTER END-OF-CODE-EXIT TO PROCEED TO ERROR-PRO
017800-    C.
017900 END-OF-CODE-EXIT.
018000     GO TO.
018100
018200 TEST-TRANS-CODE.
018300     IF OUTAREA-SW IS EQUAL TO SPACES
018400     MOVE MASTER-IN TO MASTER-OUT
018500     MOVE CURR-MASTIN-KEY TO CURR-MASTOUT-KEY
018600     PERFORM GET-MASTER-ROUTINE THROUGH GET-MASTER-EXIT.
018700     GO TO DELETE-TRANS, ADD-TRANS, CHANGE-NAME, CHANGE-ADDRESS DE
018800-    PENDING ON TC1.
018900
019000     NOTE  **TRANSACTION CODE IN ERROR.
019100
019200 TRANSACTION-CODE-ERROR.
019300     MOVE 'TRANSACTION CODE ERROR' TO ERROR-TYPE.
           MOVE HIGH-VALUE TO TRANS-CODE.
019400 ERROR-PROC.
019500     ENTER LINKAGE.
019600     CALL 'ERROR' USING TRANS-SORT-KEY, ERROR-TYPE.
019700     ENTER COBOL.
019800     GO TO READ-NEXT-TRANS.
019900 CHANGE-ADDRESS.
020000     IF OUTAREA-SW IS EQUAL TO SPACES GO TO CHANGE-ERROR.
020100     IF TC2 EQUALS 2 GO TO CHANGE-ZIP.
020300     IF TC2 IS LESS THAN 4 GO TO CHANGE-ZA, ELSE
020400     IF NOT EQUAL TO 6 GO TO TRANSACTION-CODE-ERROR.
020500     IF EQUAL TO 4 GO TO CHANGE-CITY.
020600     MOVE TRSTATE TO OUTSTATE.
020700 CHANGE-CITY.
020800     MOVE TRCITY TO OUTCITY.
020900     GO TO CHANGE-ZA.
021000 CHANGE-ZA.
021100     MOVE TRZIP TO OUTZIP.
021300     MOVE TRADDRESS TO OUTADDRESS.
021400     GO TO READ-NEXT-TRANS.
021500 CHANGE-ZIP.
021600     MOVE TRZIP TO OUTZIP.
021700     GO TO READ-NEXT-TRANS.
021800 PUT-MASTER.
021900      WRITE MASTER-OUT, MOVE SPACES TO OUTAREA-SW.
022000     IF TAK IS LESS THAN CMAK
022100      GO TO PUT-MASTER-EXIT,
022200      ELSE MOVE MASTER-IN TO MASTER-OUT, MOVE CURR-MASTIN
022300     TO CURR-MASTOUT, MOVE 'F' TO OUTAREA-SW.
           MOVE HIGH-VALUE TO CURR-MASTOUT-KEY.
022400     IF EOF-SW IS NOT EQUAL TO 'M' GO TO GET-MASTER-ROUTINE.
022500 PUT-MASTER-EXIT.
022600     EXIT.
023200 GET-MASTER-ROUTINE.
023300     NOTE.  NEW MASTER READ ROUTINE.
023400 GET-MASTER-ROUTINE-1.
023500     READ MASTER-IN,
023600         AT END GO TO MASTER-EOF.
023700     MOVE MASTIN-KEY TO CURR-MASTIN.
           MOVE HIGH-VALUE TO CURR-MASTIN-CODE.
023800     IF CURR-MASTIN IS LESS THAN PREV-MASTIN DISPLAY 'MASTER FILE
023900-    'OUT OF SEQ' UPON CONSOLE, STOP RUN.
024000     MOVE CURR-MASTIN TO PREV-MASTIN.
024100 GET-MASTER-EXIT.
024200     EXIT.
024300 GET-MASTER-EXI.
024400     GO TO PUT-MASTER-EXIT.
024500 MASTER-EOF.
024600     MOVE 'M' TO EOF-SW, MOVE HIGH-VALUE TO CURR-MASTIN.
024700      MOVE 2 TO INTC1, GO TO GET-MASTER-EXIT.
024800 END-TO-EOJ2.
024900     ALTER END-OF-ACCT-EXIT TO PROCEED TO END-TO-EOJ2-EXIT.
025000 END-OF-ACCT-ROUTINE.
025100     IF OUTAREA-SW IS NOT EQUAL TO SPACES, WRITE MASTER-OUT
025200     MOVE SPACES TO OUTAREA-SW, MASTOUT-KEY, MASTOUT-NAME-ADDRESS.
025300     CURR-MASTOUT.
025400     IF TC1 IS NOT EQUAL TO 2 MOVE 'NON MATCHING TRANSACTION'
025500     TO ERROR-TYPE AND ALTER END-OF-ACCT-EXIT TO PROCEED TO
025600     ERROR-PROC, ELSE MOVE SPACES TO ERROR-TYPE.
025700 END-OF-ACCT-EXIT.
025800     GO TO.
025900 END-TO-EOJ2-EXIT.
026000     EXIT.
026100
026200 ADD-TRANS.
026300     MOVE TRANSIN TO MASTOUT.
026400     IF TC2 IS EQUAL TO 2 AND OUTCODE IS EQUAL TO 2, MOVE 1 TO
026500      OUTCODE, WRITE MASTER-OUT, MOVE 2 TO OUTCODE.
026600     GO TO READ-NEXT-TRANS.
026700
026800 CHANGE-NAME.
026900     IF OUTAREA-SW IS EQUAL TO SPACE GO TO CHANGE-ERROR.
027000     MOVE TRNAME TO OUTNAME.
           MOVE HIGH-VALUE TO PREV-MASTOUT-KEY.
027100     GO TO READ-NEXT-TRANS.
```

B-48

PG.BX	NAME	PG.BX	NAME	PG.BX	NAME
5.13	ADD-TRANS	3.03	END-OF-CODE-ROUTINE	2.14	MOVE-KEY
3.20	CHANGE-ADDRESS	2.20	END-OF-JOB	5.23	NOTE
3.26	CHANGE-CITY	5.02	END-TO-EOJ2	4.01	PUT-MASTER
6.04	CHANGE-ERROR	5.12	END-TO-EOJ2-EXIT	4.07	PUT-MASTER-EXIT
5.20	CHANGE-NAME	2.27	EOJ1	2.05	READ-NEXT-TRANS
3.28	CHANGE-ZA	2.28	EOJ2	2.01	START
3.27	CHANGE-ZIP	3.17	ERROR-PROC	2.15	TEST-OUTAREA
2.23	COMPARE-INPUTS	4.17	GET-MASTER-EXIT	3.11	TEST-TRANS-CODE
5.22	DELETE-TRANS	5.01	GET-MASTER-EX1	2.31	TRAANS-LOW
5.10	END-OF-ACCT-EXIT	4.08	GET-MASTER-ROUTINE	2.11	TRANS-EOF
5.03	END-OF-ACCT-ROUTINE	4.09	GET-MASTER-ROUTINE-1	2.30	TRANS-LOW
3.09	END-OF-CODE-EXIT	4.15	MASTER-EOF	3.16	TRANSACTION-CODE-ERROR

COBOL MODULE STANDARD COBOL

CHART TITLE - REMARKS

CHART TITLE - PROCEDURE DIVISION

CARD ID	PAGE/BOX	NAME	REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)							
012500	2.01	START								
013000	2.05	READ-NEXT-TRANS	019800 027100	3.19 5.21	021700 027230	3.27 5.22	021400	3.28	026600	5.19
013800	2.11	TRANS-EOF	013200	2.06						
014400	2.14	MOVE-KEY	013400	2.08						
014600	2.15	TEST-OUTAREA	016100	2.26						
015400	2.18		015500	2.19						
027700	2.20	END-OF-JOB	014000	2.12	027233	6.01				
015700	2.23	COMPARE-INPUTS	014700	2.15						
015700	2.23		015400	2.18						
028100	2.27	EOJ1	028000	2.22						
028100	2.27		027800	2.21						
028300	2.28	EOJ2								
016200	2.30	TRANS-LOW	015800	2.23						
016600	2.31	TRAANS-LOW								
016800	2.32	016800	025700	5.10						
017000	3.01		016700	2.31						
017200	3.02	017200	017900	3.09						
017300	3.03	END-OF-CODE-ROUTINE								
017600	3.06		017400	3.03						
017700	3.08	017700	018000	3.09						
017900	3.09	END-OF-CODE-EXIT	017200	3.02	017700	3.08				
017900	3.09		017600	3.06						
018030	3.10	0180000	018000	3.10						
018200	3.11	TEST-TRANS-CODE	014900 025700	2.16 5.10	016000	2.25	017000	3.01	017900	3.09
018700	3.14		018300	3.11						
019200	3.15	TRANSACTION-CODE-ERROR	020400	3.23						
019400	3.17	ERROR-PROC	018000	3.09	025800	5.10	027500	6.04		
019900	3.20	CHANGE-ADDRESS	018700	3.14						
020700	3.26	CHANGE-CITY	020500	3.24						
021500	3.27	CHANGE-ZIP	020100	3.21						
021000	3.28	CHANGE-ZA	020300	3.22	020900	3.26				
021800	4.01	PUT-MASTER	015400	2.19	027900	2.22	C15900	2.24		
022500	4.07	PUT-MASTER-EXIT	015400 024400	2.19 5.01	027900	2.22	015900	2.24	022000	4.03
023200	4.08	GET-MASTER-ROUTINE	012900	2.04	018600	3.13	072400	4.06	022600	4.07
073400	4.09	GET-MASTER-ROUTINE-1								
024500	4.15	MASTER-EOF	023600	4.10						
024000	4.16		023800	4.12						
024100	4.17	GET-MASTER-EXIT	012900	2.04	019600	3.13	024700	4.15		
024300	5.01	GET-MASTER-EXI	024200	4.17						
024800	5.02	END-TO-EOJ2	028200	2.27						
024900	5.02	024900	025700	5.10						
025000	5.03	END-OF-ACCT-ROUTINE	016900	2.32						
025400	5.04		025100	5.03						
025500	5.08	025500	025800	5.10						
025600	5.09		025400	5.06						
025700	5.10	END-OF-ACCT-EXIT	016800	2.32	024900	5.02	025500	5.08		
075700	5.10		025600	5.08						
075800	5.11	0258000	025800	5.11						
025900	5.12	END-TO-EOJ2-EXIT	028200	2.27	025700	5.10				
026200	5.13	ADD-TRANS	018700	3.14	026000	5.12				
026600	5.19		026400	5.14	026400	5.15				
026800	5.20	CHANGE-NAME	019700	3.14						
027210	5.22	DELETE-TRANS	018700	3.14						
027231	5.23	NOTE								
027233	6.01	027233	027800	2.20						
027300	6.04	CHANGE-ERROR	020000	3.20	026900	5.20				

CHART TITLE - REMARKS

REMARKS. AN AUTOFLOW CHART PRODUCED FROM COBOL SOURCE INPUT MAY CONTAIN UP TO FIVE PARTS. THE LISTING OF THE SOURCE INPUT, THE TABLE OF CONTENTS AND CROSS REFERENCE LISTING, THE TABLE OF DIAGNOSTICS, A 'REMARKS' PORTION AND THE FLOW CHART PRODUCED FROM THE PROCEDURE DIVISION OF THE SOURCE PROGRAM.

THE LISTING OF THE SOURCE INPUT IS OPTIONAL, SELECTED BY THE USER IN HIS PARAMETER CARD. THE INPUT LIST OPTION HAS NOT BEEN INVOKED FOR THE AUTOFLOW CHARTING OF THIS PROGRAM.

THE TABLE OF CONTENTS INDICATES THE CHART LOCATION OF EACH SECTION OR PARAGRAPH WITHIN THE PROGRAM AS WELL AS ALL REFERENCES TO THE SECTION OR PARAGRAPH PRODUCED FROM THE GO TO, PERFORM, ALTER, AND PROCESS VERBS. IT ALSO INDICATES THE SOURCE CARD NUMBER (IN THE NAME FIELD) OF ALTER STATEMENTS ALONG WITH THE CHART LOCATION OF THE STATEMENT BEING ALTERED. CROSS REFERENCES PRODUCED BY LOCAL DECISIONS, THOSE DECISIONS WHOSE PATHS COME TOGETHER AT 'NEXT SENTENCE', ARE INDICATED WITH A BLANK NAME FIELD.

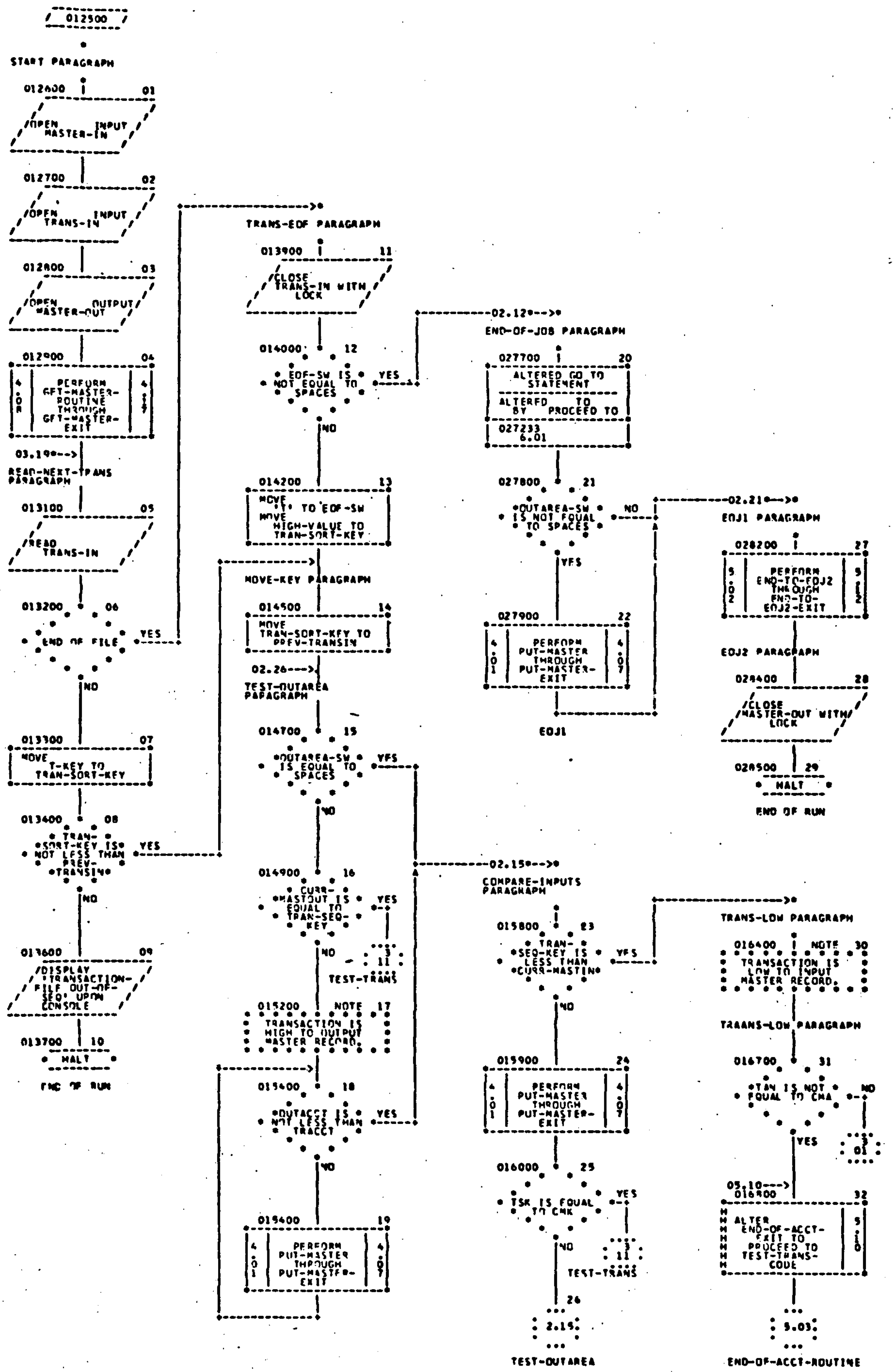
WHEN PRESENT, THE 'REMARKS' STATEMENT IN THE IDENTIFICATION DIVISION WILL APPEAR AS INTRODUCTORY NARRATIVE, IN THE TEXT FORMAT OF AN AUTOFLOW FLOW CHART. IF NO 'REMARKS' STATEMENT IS PRESENT, THIS NARRATIVE CHART IS NOT PRESENT. THE MATERIAL CURRENTLY BEING PRINTED IS AN EXAMPLE OF THE 'REMARKS' STATEMENT NARRATIVE CHART.

EACH PROCEDURE DIVISION 'SECTION' PRODUCES A CHART, WHOSE CHART TITLE IS THE SECTION NAME. THIS CHART TITLE APPEARS IN THE HEADING OF EACH FLOW CHART PAGE PRODUCED FOR THIS SECTION'S PROCEDURES. AS EACH SECTION BECOMES A CHART, THE FLOW CHARTS FOR ANY SECTION MAY BE LOCATED FROM THE TABLE OF CONTENTS.

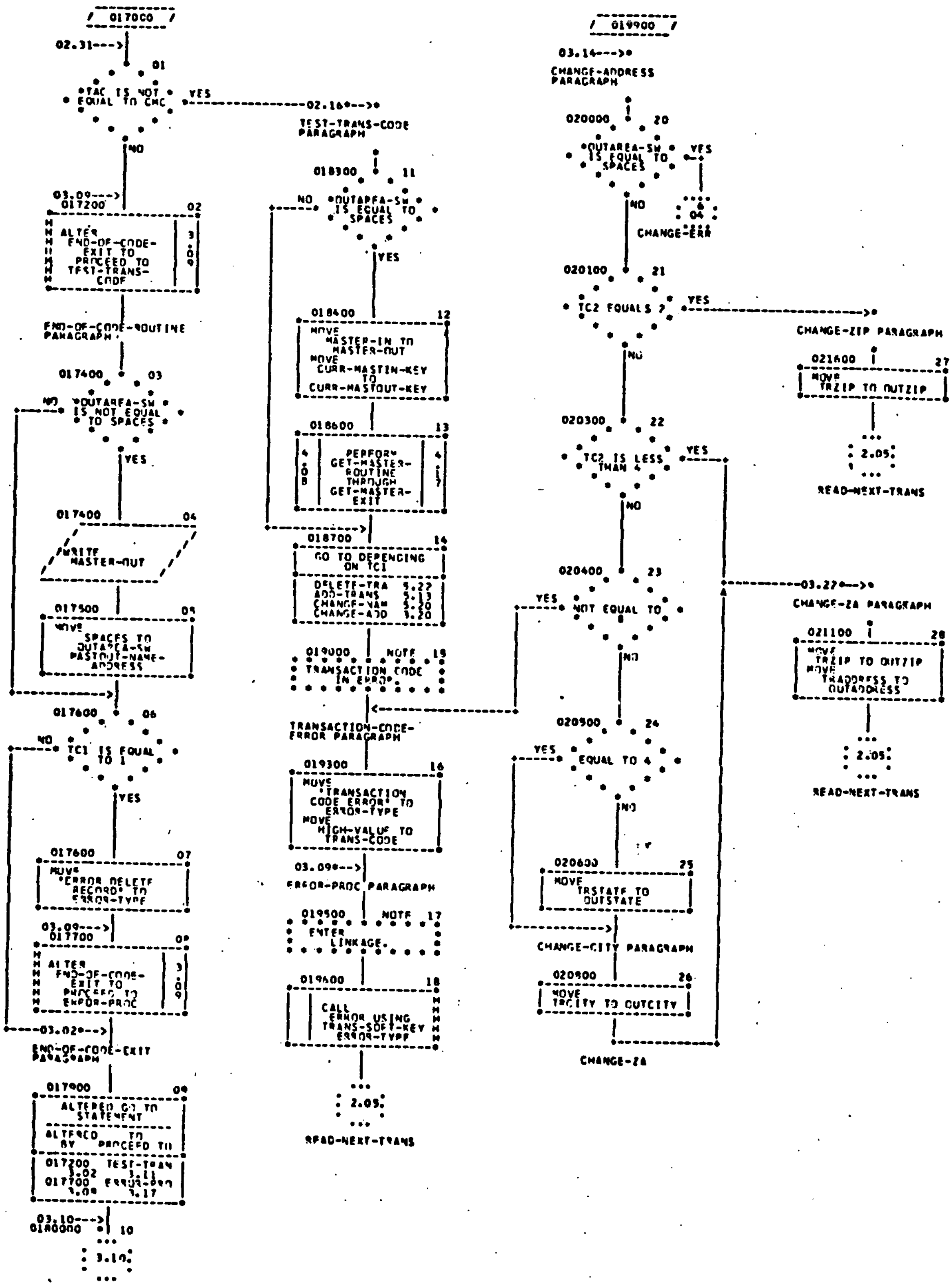
WHEN A PROGRAM IMPLIES A BRANCH FROM ONE SECTION TO THE NEXT, A BRANCH SYMBOL IS GENERATED IN THE FLOW CHART.

THIS PROGRAM IS INTENDED ONLY AS A SAMPLE OF A PROGRAM WHICH MIGHT BE RUN UNDER AUTOFLOW. IT PROBABLY IS NOT SUBJECT TO CLOSE SCRUTINY FOR LOGICAL PROGRAM CAPABILITIES.

CHART TITLE - PROCEDURE DIVISION



B-52



017000
02.31
TAC IS NOT EQUAL TO CHC
YES
02.16
TEST-TRANS-CODE PARAGRAPH
NO
03.09
017200
ALTER END-OF-CODE-EXIT TO PROCEED TO TEST-TRANS-CODE
END-OF-CODE-ROUTINE PARAGRAPH
017400
OUTAREA-SW IS NOT EQUAL TO SPACES
WRITE MASTER-OUT
017500
MOVE SPACES TO OUTAREA-SW MASTOUT-NAME-ADDRESS
017600
TC1 IS EQUAL TO 1
MOVE 'ERROR DELETE RECORD' TO ERROR-TYPE
017700
ALTER END-OF-CODE-EXIT TO PROCEED TO ERROR-PROC
03.02
END-OF-CODE-EXIT PARAGRAPH
017900
ALTERED GO TO STATEMENT
ALTERED BY PROCEED TO
018000
3.10
018300
OUTAREA-SW IS EQUAL TO SPACES
018400
MOVE MASTER-IN TO MASTER-OUT MOVE CURR-MASTIN-KEY TO CURR-MASTOUT-KEY
018600
PERFORM GET-MASTER-ROUTINE THROUGH GET-MASTER-EXIT
018700
GO TO DEPENDING ON TC1
DELETE-TRA 5.22
ADD-TRANS 5.13
CHANGE-NAM 3.20
CHANGE-ADD 3.20
019000 NOTE
TRANSACTION CODE IN ERROR.
TRANSACTION-CODE-ERROR PARAGRAPH
019300
MOVE 'TRANSACTION CODE ERROR' TO ERROR-TYPE MOVE HIGH-VALUE TO TRANS-CODE
03.09
ERROR-PROC PARAGRAPH
019500 NOTE
ENTER LINKAGE.
019600
CALL ERROR USING TRANS-SORT-KEY ERROR-TYPE
2.05
READ-NEXT-TRANS
019900
03.14
CHANGE-ADDRESS PARAGRAPH
020000
OUTAREA-SW IS EQUAL TO SPACES
CHANGE-ERR
020100
TC2 EQUALS 2
CHANGE-ZIP PARAGRAPH
021600
MOVE TRZIP TO OUTZIP
READ-NEXT-TRANS
020300
TC2 IS LESS THAN 4
020400
NOT EQUAL TO 8
03.22
CHANGE-ZA PARAGRAPH
021100
MOVE TRZIP TO OUTZIP MOVE TRADDRESS TO OUTADDRESS
020500
EQUAL TO 4
020600
MOVE TRSTATE TO OUTSTATE
CHANGE-CITY PARAGRAPH
020800
MOVE TRCITY TO OUTCITY
CHANGE-ZA

```
04/27/70      PROCEDURE DIVISION SUMMARY                AUTOFLOW CHART SET - SAMPLE   STANDARD COBOL                    PAGE   1

ALTERED PARAGRAPHS

   PARAGRAPH    0025700   5.10     END-OF-ACCT-EXIT                 INITIALLY                                   UNSPECIFIED
   ALTERED AT   0015800   2.32 IN  TRAANS-LOW                        TO PROCEED TO 0019200    3.11              TEST-TRANS-CODE
                0024900   5.07 IN  END-TO-EOJ2                                     0025900    5.12              END-TO-EOJ2-EXIT
                0025500   5.08 IN  END-OF-ACCT-ROUTINE                             0019400    3.17              ERROR-PROC
   PARAGRAPH    0017900   3.09     END-OF-CODE-EXIT                 INITIALLY                                   UNSPECIFIED
   ALTERED AT   0017200   3.02 IN  TRAANS-LOW                        TO PROCEED TO 0018200    3.11              TEST-TRANS-CODE
                0017700   3.08 IN  END-OF-CODE-ROUTINE                             0019400    3.17              ERROR-PROC
   PARAGRAPH    0027700   2.20     END-OF-JOB                       INITIALLY                                   UNSPECIFIED
   ALTERED AT   0027233   6.01 IN  NOTE                              TO PROCEED TO                              ERROR-2
------------------------------------------------------------------------------------------------------------------------------
PERFORMED PROCEDURES
 END-TO-EOJ2
   PERFORMED AT  0028200   2.27 IN EOJ1                         PERFORM ENTRY 0024800   5.02
                                                                        THRU  0025900   5.12     END-TO-EOJ2-EXIT
 GET-MASTER-ROUTINE
   PERFORMED AT  0012900   2.04 IN START                        PERFORM ENTRY 0023200   4.08
                                                                        THRU  0024100   4.17     GET-MASTER-EXIT
                 0018600   3.13 IN TEST-TRANS-CODE                      ENTRY 0023200   4.08
                                                                        THRU  0024100   4.17     GET-MASTER-EXIT
 PUT-MASTER
   PERFORMED AT  0015400   2.19 IN TEST-OUTAREA                 PERFORM ENTRY 0021800   4.01
                                                                        THRU  0022500   4.07     PUT-MASTER-EXIT
                 0015900   2.24 IN COMPARE-INPUTS                       ENTRY 0021800   4.01
                                                                        THRU  0022500   4.07     PUT-MASTER-EXIT
                 0027900   2.22 IN END-OF-JOB                           ENTRY 0021800   4.01
                                                                        THRU  0022500   4.07     PUT-MASTER-EXIT
------------------------------------------------------------------------------------------------------------------------------
I/O FILE ACTIVITY
 MASTER-IN
                 0012600   2.01 IN START                        OPEN INPUT MASTER-IN
                 0023500   4.09 IN GET-MASTER-ROUTINE-1         READ MASTER-IN
 MASTER-OUT
                 0012800   2.03 IN START                        OPEN OUTPUT MASTER-OUT
                 0017400   3.04 IN END-OF-CODE-ROUTINE          WRITE MASTER-OUT
                 0021900   4.01 IN PUT-MASTER                   WRITE MASTER-OUT
                 0025100   5.04 IN END-OF-ACCT-ROUTINE          WRITE MASTER-OUT
                 0026500   5.17 IN ADD-TRANS                    WRITE MASTER-OUT
                 0028400   2.28 IN EOJ2                         CLOSE MASTER-OUT WITH LOCK
 TRANS-IN
                 0012700   2.02 IN START                        OPEN INPUT TRANS-IN
                 0013100   2.05 IN READ-NEXT-TRANS              READ TRANS-IN
                 0013900   2.11 IN TRANS-EOF                    CLOSE TRANS-IN WITH LOCK
------------------------------------------------------------------------------------------------------------------------------
STOP SUMMARY
                 0013700   2.10 IN READ-NEXT-TRANS              STOP RUN
                 0023900   4.14 IN GET-MASTER-ROUTINE-1         STOP RUN
                 0028500   2.29 IN EOJ2                         STOP RUN
------------------------------------------------------------------------------------------------------------------------------
DISPLAY SUMMARY
                 0013600   2.09 IN READ-NEXT-TRANS              DISPLAY 'TRANSACTION-FILE OUT-OF-SEQ' UPON CONSOLE
                 0023800   4.13 IN GET-MASTER-ROUTINE-1         DISPLAY 'MASTER FILE OUT OF SEQ' UPON CONSOLE
------------------------------------------------------------------------------------------------------------------------------
CALLED PROCEDURES
 ERROR
   CALLED FROM   0019600   3.18 IN ERROR-PROC
```

CARD ID	DATA NAME	REFERENCES (SEQUENCE NO. AND PAGE/ROW)							
C03400	CMA	016700	2.31						
008900	CMAK	022000	4.03						
009900	CMC	017000	3.01						
C09600	CMR	016000	2.25						
008400	CURR-MASTIN	015800	2.23	022200	4.04	023700	4.11-M	023800	4.12
		024600	4.15-M	024000	4.16				
008600	CURR-MASTIN-ACCT								
009000	CURR-MASTIN-CODE	(0002741	4.11-M						
008500	CURR-MASTIN-KEY	018500	3.12						
008900	CURR-MASTIN-REFER								
008700	CURR-MASTIN-TYPE								
009100	CURR-MASTOUT	014900	2.16	022300	4.04-M	025300	5.05-M		
009300	CURR-MASTOUT-ACCT	027220	5.22-M						
009800	CURR-MASTOUT-CODE								
009200	CURR-MASTOUT-KEY	018500	3.12-M	(0002641	4.05-M				
009700	CURR-MASTOUT-REFER								
009500	CURR-MASTOUT-TYPE								
008100	EOF-SW	014000	2.12	014200	2.13-M	022400	4.06	024600	4.15-M
008300	ERROR-TYPE	017700	3.07-M	019300	3.16-M	019600	3.18-U	025500	5.07-M
		025600	5.09-M	027400	6.04-M				
003100	INACCT								
003700	INADDRESS								
003800	INCITY								
003400	INCODE								
003600	INNAME								
003300	INREF								
003900	INSTATE								
003200	INTYPE								
004000	INZIP								
002400	MASTER-IN	012600	2.01	018400	3.12	022200	4.04	023500	4.09-M
004100	MASTER-OUT	012800	2.03	028400	2.29	017400	3.04	018400	3.12-M
		021900	4.01	022200	4.04-M	025100	5.04	026500	5.17
002900	MASTIN								
003000	MASTIN-KEY	023700	4.11						
003500	MASTIN-NAME-ADDRESS								
004600	MASTOUT	026300	5.13-M						
004700	MASTOUT-KEY	025200	5.05-M						
005200	MASTOUT-NAME-ADDRESS	017500	3.05-M	025200	5.05-M				
004800	OUTACCT	015400	2.23						
005500	OUTADDRESS	021300	3.28-M						
009700	OUTAREA-SW	014700	2.15	027800	2.20	017400	3.03	017500	3.05-M
		019300	3.11	020000	3.20	021900	4.02-M	022300	4.04-M
		025100	5.03	025200	5.05-M	026900	5.20	027220	5.22-M
005600	OUTCITY	020800	3.26-M						
005100	OUTCODE	026400	5.15	026500	5.16-M	026500	5.18-M		
005300	OUTNAME	027000	5.21-M						
005000	OUTREF								
C05700	OUTSTATE	020600	3.25-M						
004900	OUTYPE								
005800	OUTZIP	021600	3.27-M	021100	3.28-M				
007800	PREV-MASTIN	023800	4.12	024000	4.16-M				
007900	PREV-MASTOUT								
010000	PREV-MASTOUT								
010200	PREV-MASTOUT-ACCT								
010500	PREV-MASTOUT-CODE								
010100	PREV-MASTOUT-KEY	(0003081	5.21-M						
010400	PREV-MASTOUT-REFER								
010300	PREV-MASTOUT-TYPE								
008000	PREV-TRANSIN	013400	2.08	014500	2.14-M				
011600	TAC	017000	3.01						
011500	TACCT-CODE								
011100	TACCT-NO								
011300	TACCT-TYPE								
011350	TAK	022000	4.03						
012200	TAMT								
011200	TAN	016700	2.31						
011900	TC1	017600	3.06	018800	3.14	025400	5.06		
012000	TC2	020100	3.21	020300	3.22	026400	5.14		
006600	TRACCT	015500	2.23						
007200	TRADDRESS	021300	3.28						
010800	TRAN-ACCT								
010900	TRAN-ACCT-KEY								
C10700	TRAN-SEQ-KEY	014900	2.16	015800	2.23				
010600	TRAN-SORT-KEY	013300	2.07-M	013400	2.08	014300	2.13-M	014500	2.14
011800	TRANS-CODE	(0002351	3.16-M						
005900	TRANS-IN	012700	2.02	013100	2.05-M	013900	2.11		
007000	TRANS-NAME-ADDRESS								
C06400	TRANSIN	026300	5.13						
006500	TRANSIN-KEY								

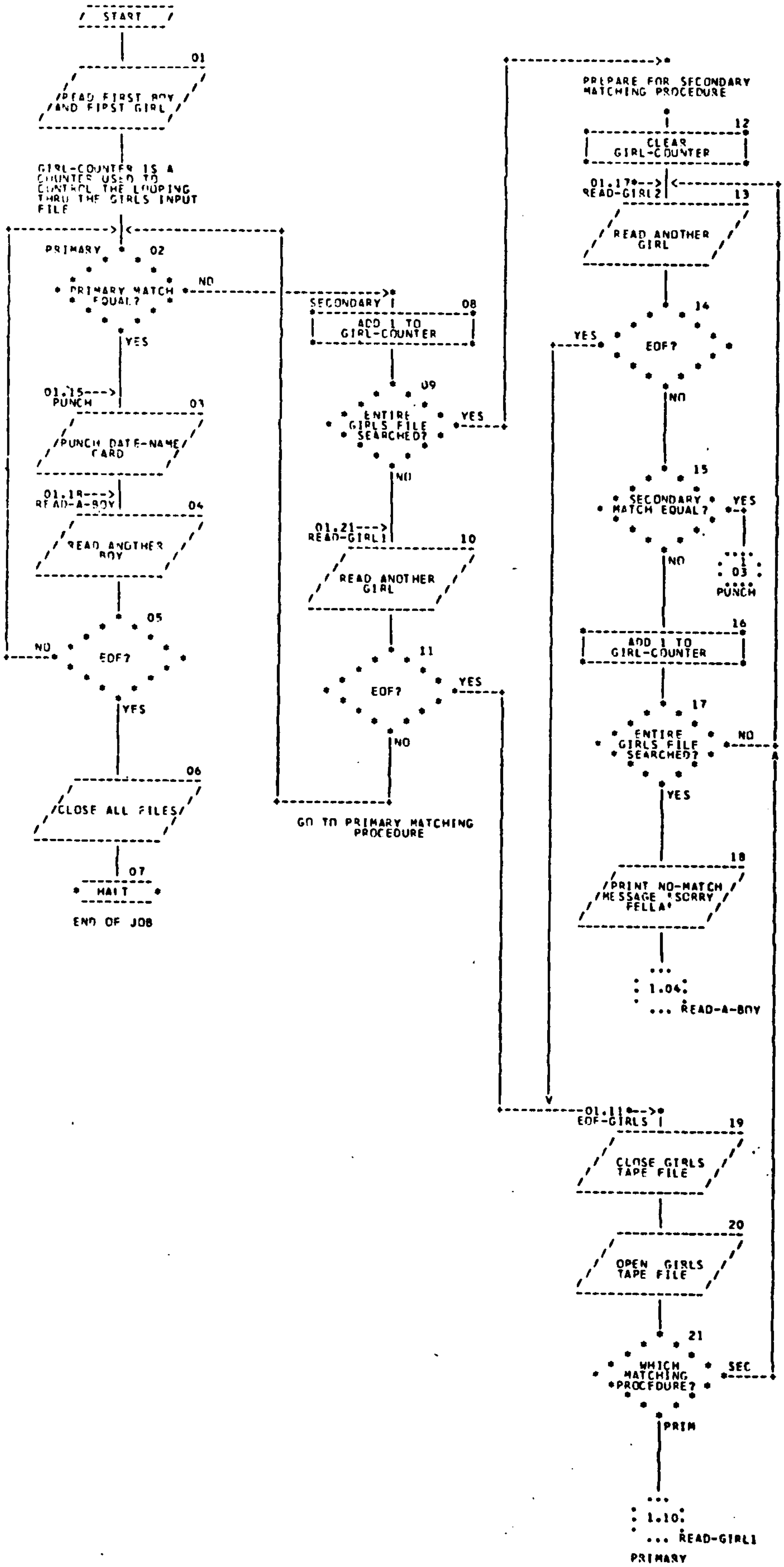
```
04/27/70          DATA RECORD MAP                    AUTOFLOW CHART SET - SAMPLE   STANDARD COBOL                    PAGE  1

FILE SECTION
   ID        LVL      DATA NAME                       OFFSET    LENGTH   # DEC   OCCURS   CLASS    USAGE     RB VALUE

002400    FD  MASTER-IN                  LABEL RECORD IS STANDARD, BLOCK CONTAINS 2000 CHARACTERS, RECORD CONTAINS 100
                                          CHARACTERS
002900     01  MASTIN
003000      02  MASTIN-KEY
003100       03  INACCT                          0-9       10                         A/N     DISPLAY
003200       03  INTYPE                         10-11       2                         A/N     DISPLAY
003300       03  INREF                          12-27      16                         A/N     DISPLAY
003400       03  INCODE                         28-28       1                         A/N     DISPLAY
003500      02  MASTIN-NAME-ADDRESS
003600       03  INNAME                         29-54      26                         A/N     DISPLAY
003700       03  INADDRESS                      55-77      23                         A/N     DISPLAY
003800       03  INCITY                         78-92      15                         A/N     DISPLAY
003900       03  INSTATE                        93-94       2                         A/N     DISPLAY
004000       03  INZIP                          95-99       5                         A/N     DISPLAY
004100    FD  MASTER-OUT                 LABEL RECORD IS STANDARD, BLOCK CONTAINS 2000 CHARACTERS, RECORD CONTAINS 100
                                          CHARACTERS
004600     01  MASTOUT
004700      02  MASTOUT-KEY
004800       03  OUTACCT                         0-9       10                         A/N     DISPLAY
004900       03  OUTYPE                         10-11       2                         A/N     DISPLAY
005000       03  OUTREF                         12-27      16                         A/N     DISPLAY
005100       03  OUTCODE                        28-28                                 A/N     DISPLAY
005200      02  MASTOUT-NAME-ADDRESS
005300       03  OUTNAME                        29-54      26                         A/N     DISPLAY
005500       03  OUTADDRESS                     55-77      23                         A/N     DISPLAY
005600       03  OUTCITY                        78-92      15                         A/N     DISPLAY
005700       03  OUTSTATE                       93-94       2                         A/N     DISPLAY
005900       03  OUTZIP                         95-99       5                         A/N     DISPLAY
005900    FD  TRANS-IN                   LABEL RECORD IS STANDARD, BLOCK CONTAINS 2000 CHARACTERS, RECORD CONTAINS 100
                                          CHARACTERS
006400     01  TRANSIN
006500      02  TRANSIN-KEY
006600       03  TRACCT                          0-9       10                         A/N     DISPLAY
006700       03  TRTYPE                         10-11       2                         A/N     DISPLAY
006800       03  TREF                           12-27      16                         A/N     DISPLAY
006900       03  TRCODE                         28-28       1                         A/N     DISPLAY
007000      02  TRANS-NAME-ADDRESS
007100       03  TRNAME                         29-54      26                         A/N     DISPLAY
007200       03  TRADDRESS                      55-77      23                         A/N     DISPLAY
007300       03  TRCITY                         78-92      15                         A/N     DISPLAY
007400       03  TRSTATE                        93-94       2                         A/N     DISPLAY
007500       03  TRZIP                          95-99       5                         A/N     DISPLAY
```

04/27/73   DATA DIVISION INDEX   AUTOFLOW CHART SET - SAMPLE   STANDARD COBOL   PAGE 1

ID	LVL	DATA NAME	OFFSET	RECORD	FILE	NO VALUE
009400	03	CMA	186-191	CURR-MASTOUT	WORKING-STORAGE SECTION	
[illegible]	03	CMAN	152-153	CURR-MASTIN	..	
009900	02	CMC	212-212	CURR-MASTOUT	..	
[illegible]	02	CME	186-195	..	..	
[illegible]	01	CURR-MASTIN			..	
[illegible]	03	CURR-MASTIN-ACCT	152-161	CURR-MASTIN	..	
[illegible]	02	CURR-MASTIN-CODE	180-180	..	..	
[illegible]	02	CURR-MASTIN-KEY		..	..	
[illegible]	02	CURR-MASTIN-REFER	164-179	..	..	
[illegible]	03	CURR-MASTIN-TYPE	162-163	..	..	
[illegible]	01	CURR-MASTOUT			..	
009300	03	CURR-MASTOUT-ACCT	186-193	CURR-MASTOUT	..	
[illegible]	02	CURR-MASTOUT-CODE	212-212	..	..	
009200	02	CURR-MASTOUT-KEY		..	..	
[illegible]	02	CURR-MASTOUT-REFER	196-211	..	..	
[illegible]	03	CURR-MASTOUT-TYPE	194-195	..	..	
008100	77	EOF-SW	108-108		..	
008300	77	ERROR-TYPE	110-149		..	
003100	03	INACCT	0-9	MASTIN	MASTER-IN	
003700	03	INADDRESS	55-77	..	..	
[illegible]	03	INCITY	78-92	..	..	
003400	03	INCODE	28-28	..	..	
003600	03	INNAME	29-54	..	..	
003300	03	INREF	12-27	..	..	
003900	03	INSTATE	94-96	..	..	
003200	03	INTYPE	10-11	..	..	
004000	03	INZIP	95-99	..	..	
002400	FD	MASTER-IN				
004100	FD	MASTER-OUT				
002900	01	MASTIN			MASTER-IN	
003000	02	MASTIN-KEY		MASTIN	..	
003500	02	MASTIN-NAME-ADDRESS		..		
004400	01	MASTOUT			MASTER-OUT	
004700	02	MASTOUT-KEY		MASTOUT	..	
005200	02	MASTOUT-NAME-ADDRESS		..	..	
004800	03	OUTACCT	0-9	..	..	
005500	03	OUTADDRESS	55-77	..	..	
009200	77	OUTAREA-SW	109-109		WORKING-STORAGE SECTION	
005600	03	OUTCITY	78-92	MASTOUT	MASTER-OUT	
005100	03	OUTCODE	28-28	..	..	
005300	03	OUTNAME	29-54	..	..	
005000	03	OUTREF	12-27	..	..	
005700	03	OUTSTATE	93-94	..	..	
004900	03	OUTTYPE	10-11	..	..	
005800	03	OUTZIP	95-99	..	..	
007800	77	PREV-MASTIN	0-35		WORKING-STORAGE SECTION	
010000	01	PREV-MASTOUT			..	
007900	77	PREV-MASTOUT	36-71		..	
010200	03	PREV-MASTOUT-ACCT	210-225	PREV-MASTOUT	..	
010500	02	PREV-MASTOUT-CODE	244-244	..	..	
010100	02	PREV-MASTOUT-KEY		..	..	
010400	02	PREV-MASTOUT-REFER	228-243	..	..	
010300	03	PREV-MASTOUT-TYPE	226-227	..	..	
008000	77	PREV-TRANSIN	72-107		..	
011600	03	TAC	276-276	TRAN-SORT-KEY	..	
011500	03	TACCT-CODE	276-276	..	..	
011100	05	TACCT-NO	249-257	..	..	
011300	05	TACCT-TYPE	258-259	..	..	
011350	04	TAK	248-259	..	..	
017200	02	TAMT	279-285	..	..	
011200	05	TAN	249-257	..	..	
011900	03	TC1	277-277	..	..	
012000	03	TC2	278-278	..	..	
006600	03	TRACCT	0-9	TRANSIN	TRANS-IN	
007700	03	TRADDRESS	55-77	..	..	
[illegible]	03	TRAN-ACCT		TRAN-SORT-KEY	WORKING-STORAGE SECTION	
010900	04	TRAN-ACCT-KEY		..	..	
010700	02	TRAN-SEQ-KEY		..	..	
010600	01	TRAN-SORT-KEY			..	
011800	02	TRANS-CODE		TRAN-SORT-KEY	..	
005900	FD	TRANS-IN				
007000	02	TRANS-NAME-ADDRESS		TRANSIN	TRANS-IN	
006400	01	TRANSIN			..	
006500	02	TRANSIN-KEY		TRANSIN	..	
007200	03	TRCITY	78-92	..	..	
006900	03	TRCODE	28-28	..	..	
[illegible]	03	TREF	12-27	..	..	
011400	03	TREFERENCE	260-275	TRAN-SORT-KEY	WORKING-STORAGE SECTION	
007100	03	TRNAME	29-54	TRANSIN	TRANS-IN	
007400	03	TRSTATE	93-94	..	..	
006700	03	TRTYPE	10-11	..	..	
007500	03	TRZIP	95-99	..	..	
011700	02	TSK	248-276	TRAN-SORT-KEY	WORKING-STORAGE SECTION	

```
04/27/70        INPUT LISTING                          AUTOFLOW CHART SET - SAMPLE   CHART COBOL FACILITIES
CHART/COBOL -        (LIST2)
 CARD NO        ****                        CONTENTS                                ****
      1              PROCEDURE DIVISION.
      2                  NOTE
      3                  R  THE MATCHING OF BOYS & GIRLS FOR HIGH SCHOOL DANCE..
      4              START.
      5                  NOTE
      6                  I  READ FIRST BOY AND FIRST GIRL.
      7                  T  GIRL-COUNTER IS A COUNTER USED TO CONTROL THE LOOPING
      8                    THRU THE GIRLS INPUT FILE.
      9              PRIMARY.
     10                  NOTE
     11                  DY  (SECONDARY) PRIMARY MATCH EQUAL?.
     12                  I  .PUNCH PUNCH DATE-NAME CARD.
     13              READ-A-BOY.
     14                  NOTE
     15                  I  READ ANOTHER BOY.
     16                  DY  (PRIMARY) EOF?.
     17                  I  CLOSE ALL FILES.
     18                  H  END OF JOB.
     19              SECONDARY.
     20                  NOTE
     21                  P  ADD 1 TO GIRL-COUNTER.
     22                  DN  (PREPARE) ENTIRE GIRLS FILE SEARCHED?.
     23              READ-GIRL1.
     24                  NOTE
     25                  I  READ ANOTHER GIRL.
     26                  DN  (EOF-GIRLS) EOF?.
     27                  B  (PRIMARY) GO TO PRIMARY MATCHING PROCEDURE.
     28              PREPARE.
     29                  NOTE
     30                  T  PREPARE FOR SECONDARY MATCHING PROCEDURE.
     31                  P  CLEAR GIRL-COUNTER.
     32              READ-GIRL2.
     33                  NOTE
     34                  I  READ ANOTHER GIRL.
     35                  DN  (EOF-GIRLS) EOF?.
     36                  DN  (PUNCH) SECONDARY MATCH EQUAL?.
     37                  P  ADD 1 TO GIRL-COUNTER.
     38                  DY  (READ-GIRL2) ENTIRE GIRLS FILE SEARCHED?.
     39                  I  PRINT NO-MATCH MESSAGE 'SORRY FELLA'.
     40                  B  (READ-A-BOY).
     41              EOF-GIRLS.
     42                  NOTE
     43                  I  CLOSE GIRLS TAPE FILE.
     44                  I  OPEN  GIRLS TAPE FILE.
     45                  D  (PRIM,SEC-READ-GIRL2) WHICH MATCHING PROCEDURE?.
     46                  B  (READ-GIRL1) PRIMARY.
```

CHART TITLE - THE MATCHING OF BOYS & GIRLS FOR HIGH SCHOOL DANCE.



ASSEMBLY MODULE (LIST1.NAMSO)

```
CARD NO  ****                          CONTENTS                                        ****

                  OPTION MODACR=YES,DCS=YES,MACMAP=YES,LITSUM=YES
         *        LPRT     I                                                    *D   IV010140
         *        IFERR    DN                                                   *D   IV010150
         IV01     TITLE 'VERIFY INVENTORY SYSTEM UPDATES'                            IV010160
                  MACRO                                                              IV010170
         &NAME    LPRT   &RECORD,&ADV,&LENGTH                                        IV010180
                  LCLC   &FUDGE                                                      IV010190
         &FUDGE   SETC   'L'''                                                       IV010200
         &NAME    MVI    PRBUF+1,C' '              CLEAR PRINT BUFFER                IV010210
                  ST     15,SYSREG15
                  MVC    PRBUF+2(&FUDGE.PRBUF-2),PRBUF+1                             IV010220
                  AIF    (T'&LENGTH EQ 'O').MLA                                      IV010230
                  MVC    PRBUF+1(&LENGTH),&RECORD      MOVE # OF BYTES SPECIFIED     IV010240
                  AGO    .ADV                                                        IV010250
         .MLA     ANOP                                                               IV010260
                  MVC    PRBUF+1(&FUDGE&RECORD),&RECORD    MOVE ACCORDING TO L'      IV010270
         .ADV     ANOP                                                               IV010280
                  AIF    ('&ADV' EQ '1').ADV1                                        IV010290
                  AIF    ('&ADV' EQ '2').ADV2                                        IV010300
                  AIF    ('&ADV' EQ '3').ADV3                                        IV010310
                  AIF    ('&ADV' EQ 'E').ADVEJ                                       IV010320
                  AIF    ('&ADV' EQ 'EJECT').ADVEJ                                   IV010330
                  MNOTE  255,'INVALID PRINTER ADVANCE SPECIFIED'                     IV010340
                  MEXIT                                                              IV010350
         .ADV1    MVI    PRBUF,X'09'                                                 IV010360
                  AGO    .COMMON                                                     IV010370
         .ADV2    MVI    PRBUF,X'11'                                                 IV010380
                  AGO    .COMMON                                                     IV010390
         .ADV3    MVI    PRBUF,X'19'                                                 IV010400
                  AGO    .COMMON                                                     IV010410
         .ADVEJ   MVI    PRBUF,X'89'                                                 IV010420
         .COMMON  ANOP                                                               IV010430
                  L      13,=A(SAVEAREA)                                             IV010440
                  PUT    ERROR,PRBUF                                                 IV010450
                  L      15,SYSREG15
                  MEND                                                               IV010460
                  MACRO                                                              IV010480
         &NAME    RES    &VAR                                                        IV010490
         &NAME    MVI    &VAR,X'00'                                                  IV010500
                  MEND                                                               IV010510
                  MACRO                                                              IV010530
         &NAME    SET    &VAR                                                        IV010540
         &NAME    MVI    &VAR,X'FF'                                                  IV010550
                  MEND                                                               IV010560
                  MACRO                                                              IV010580
         &NAME    IFERR  &DEST                                                       IV010590
         &NAME    CLI    IPERRPR,X'FF'                                               IV010600
                  BE     &DEST                                                       IV010610
                  MEND                                                               IV010620
                  MACRO                                                              IV010640
         &NAME    CIEPR                                                              IV010650
         &NAME    CLI    IPERRPR,X'FF'                                               IV010660
                  BE     @&SYSNDX                                                    IV010670
                  LPRT   INAREAPR,1,120                                              IV010680
                  SET    IPERRPR                                                     IV010690
         @&SYSNDX DS     0H                                                          IV010700
                  MEND                                                               IV010710
                  EJECT                                                              IV010720
         IV01PJB  CSECT                                                              IV010730
                  USING  IV01PJB,11                                                  IV010740
         MAIN     GET    INPUT,INAREA                                                IV010760
                  RES    IPERRPR                                                     IV010770
         *        CLEAR OUTPUT RECORD                                                IV010780
                  MVC    OUTPCODE,SPACES                                             IV010790
                  RES    OUTNCODE                                                    IV010800
                  RES    OUTDCODE                                                    IV010810
                  RES    OUTCCODE                                                    IV010820
                  MVC    OUTNAME,SPACES                                              IV010840
                  ZAP    OUTVOL,=P'0'                                                IV010850
                  CLI    IACODE,C'0'                                                 IV010860
                  BL     ACERROR                                                     IV010870
                  BE     ADD                                                         IV010880
                  CLI    IACODE,C'2'                                                 IV010890
                  BL     NAMCHG                                                      IV010900
                  BE     INVINC                                                      IV010910
                  CLI    IACODE,C'4'                                                 IV010920
                  BL     INVDEC                                                      IV010930
                  BE     DEL                                                         IV010940
         ACERROR  LPRT   INAREAPR,1,120                                              IV010950
                  LPRT   MESS01,1                                                    IV010960
         INERROR  LPRT   SPACES,3                                                    IV010970
                  B      MAIN                                                        IV010980
                  EJECT
         INVINC   DS     0H
         *        PROCESSING FOR INVENTORY INCREASE/DECREASE
                  CLC    IPNAME,SPACES
                  BE     IINC05
                  CIEPR
                  LPRT   MESS11,1
         IINC05   LA     14,IINC10
                  CNOP   0,4
                  BAL    1,*+12
                  DC     A(IPVCL)
                  DC     A(VOLDEC)
         IINC07   L      15,=V(DECCVT)
         IINC10   ZAP    VOLDEC,=P'0'
                  BR     15
                  C      15,=F'4'
                  BL     IINC20
                  BH     IINC30
                  CLI    IACODE,C'3'
                  BE     IINC60
                  AP     OUTVOL,VOLDEC
         IINC50   AP     HASHTOT,=P'1'
                  BC     *+10
                  LA     3,WRITE
                  BR     3
                  ABEND  4000,DUMP
         IINC20   CIEPR
                  LPRT   MESS14,1
                  B      IINC50
         IINC30   CIEPR
                  LPRT   MESS06,1
                  B      IINC50
         IINC60   SP     OUTVOL,VOLDEC
                  B      IINC50
         INVDEC   EQU    INVINC
                  EJECT
         DEL      DS     0H                                                          IV011000
         *        PROCESSING FOR DELETE PRODUCT CODE                                 IV011010
                  CLC    IPNAME,SPACES                                               IV011020
                  BNE    DEL10                                                       IV011030
                  CIEPR                                                              IV011050
         DEL05    LPRT   MESS02,1                                                    IV011040
         DEL10    L      15,=V(DECCVT)                                               IV011060
                  CALL   (15),(IPVCL,VOLDEC)                                         IV011080
                  C      15,=F'0'                                                    IV011090
                  BE     DEL15                                                       IV011100
                  CIEPR                                                              IV011110
         DEL20    LPRT   MESS05,1                                                    IV011120
                  C      15,=F'4'                                                    IV011180
                  BE     DEL15
         DEL30    LPRT   MESS06,1                                                    IV011130
         DEL15    SET    OUTDCODE                                                    IV011190
                  BC     15,WRITE                                                    IV011200
                  EJECT
         ADD      DS     0H                                                          IV011760
         *        PROCESSING FOR ADD                                                 IV011770
                  SET    OUTNCODE                                                    IV011790
                  CLC    IPNAME,SPACES                                               IV011780
                  BE     ADD10                                                       IV011800
         ADD15    L      15,=V(DECCVT)                                               IV011830
                  CALL   (15),(IPVCL,VOLDEC)                                         IV011840
                  C      15,=F'0'                                                    IV011850
                  BE     WRITE
                  CIEPR                                                              IV011920
         ADD30    LPRT   MESS08,1                                                    IV011930
                  C      15,=F'4'
                  BE     WRITE
                  LPRT   MESS06,1                                                    IV011880
                  B      WRITE                                                       IV011940
         ADD10    CIEPR                                                              IV011810
         ADD05    LPRT   MESS07,1                                                    IV011820
```

CARD NO   ****   CONTENTS   ****

```
         B     ADO15
         EJECT                                                          IV011210
NAMCHG   DS    0H                                                       IV011960
*        PROCESSING FOR NAME CHANGE                                     IVC11970
         CLC   IPNAME,SPACES                                            IV011980
         SET   OUTCCODE                                                 IV011990
         BNE   NAM10                                                    IVC12000
         CIEPR                                                          IV012010
NAM05    LPRT  MESS09,1                                                 IV012020
NAM10    L     15,=V(DECCVT)                                            IV012030
         CALL  (15),(IPVCL,VOLDEC)                                      IV012040
         C     15,=F'0'
         BE    WRITE
         CIEPR                                                          IV012120
NAM30    LPRT  MESS10,1                                                 IV012130
         C     15,=F'4'
         BE    WRITE
         LPRT  MESS06,1                                                 IV012080
         B     WRITE                                                    IV012140
         EJECT                                                          IV012150
WRITE    MVC   OUTNAME,IPNAME                                           IV011230
         MVC   OUTPCCDE,IPCODE                                          IV011240
* RANGE TEST PRODUCT CODE                                               IV011260
* CHAR   1     A - I                                                    IV011270
* CHAR 2 - 4   NUMERIC                                                  IVC11280
* CHAR   5     0 - 5                                                    IV011290
* CHAR 6 - 7   NUMERIC                                                  IVC11300
* CHAR   8     ALPHA                                                    IV011310
         CLI   IPCODE,X'C1'                                             IV011330
         BC    4,WRITE01                                                IV011340
         CLI   IPCODE,X'C9'                                             IV011350
         BC    2,WRITE01                                                IV011360
         LM    3,5,=A(IPCCOE+1,1,IPCODE+3)                              IV011370
WRITE10  CLI   0(3),X'F0'                                               IV011380
         BL    WRITE01                                                  IV011390
         CLI   0(3),X'F9'                                               IVC11400
         BH    WRITE01                                                  IV011410
         BXLE  3,4,WRITE10                                              IV011420
         CLI   IPCODE+4,C'0'                                            IV011440
         BL    WRITE44
         CLI   IPCODE+4,C'5'                                            IV011460
         BH    WRITE11                                                  IV011470
         CLI   IPCODE+5,X'F0'                                           IV011480
         BL    WRITE01                                                  IV011490
         CLI   IPCODE+5,X'F9'                                           IV011500
         BNH   *+8
         B     WRITE33
         CLI   IPCODE+6,X'F0'                                           IV011520
         BL    WRITE01                                                  IV011530
         CLI   IPCODE+6,X'F9'                                           IV011540
WRITE33  BH    WRITE01                                                  IV011550
         TM    IPCODE+7,X'C0'
         BNC   WRITE01
         CLI   IPCODE+7,X'F0'
         BL    WRITE01
WRITE20  IFERR INERRCR                                                  IVC11720
         WRITE DECB,SF,OUTPUT,CUTAREA,100                               IV010730
         CHECK DECB                                                     IVC10735
         B     MAIN                                                     IV011740
WRITE01  IFERR WRITE02                                                  IV011680
         LPRT  INAREAPR,1,120                                           IV011690
         SET   IPERRPR                                                  IV011700
WRITE02  LPRT  MESS03,1                                                 IV011710
         B     MAIN
WRITE44  EOU   WRITE26
WRITE11  ECU   WRITE01                                                  IV011220
         EJECT
INAREAPR DC    CL40' '                                                  IV012160
INAREA   DS    0CL80                                                    IVC12170
IACODE   DC    CL1' '                 ACTIVITY CODE                     IV012180
IPCODE   DC    CL8' '                 PRODUCT CODE                      IV012190
IPNAME   DC    CL50' '                PRODUCT NAME                      IV012200
IPVOL    DC    CL10' '                PRODUCT ACTIVITY VOLUME           IV012210
         ORG   INAREA+80                                                IV012220
OUTAREA  DS    0CL100                                                   IVC12240
OUTPCODE DS    CL10' '                PRODUCT CCDE                      IV012250
CUTNCODE DC    CL1' '                 SET TO X'FF' FOR NEW PRODUCT      IV012260
OUTDCODE DC    CL1' '                 SET TO X'FF' FOR PRODUCT DELETE   IV012270
OUTCCODE DC    CL1' '                 SET TO X'FF' FOR NAME CHANGE      IV012280
OUTVOL   DC    P'0000000000'                                            IVC12290
OUTNAME  DC    CL50' '                                                  IV012300
         ORG   OUTAREA+100                                              IV012310
SPACES   DC    CL80' '                                                  IV012330
SPACE EOU SPACES
BLANK EOU SPACES
HASHTOT  DC    P'00000'
IPERRPR  DC    X'00'                                                    IV012340
SYSREG15 DS    1F
VOLDEC   DC    P'CC000C0000'                                            IV012360
PRBUF    DC    CL121' '
         EJECT                                                          IV012380
MESS01   DC    C'********* INVALID ACTIVITY CODE '                      IV012390
MESS02   DC    C'********* PRODUCT NAME NOT SPECIFIED FOR DELETE        IV012400
MESS03   DC    C'********* PRODUCT CODE NOT ACCEPTABLE'                 IV012410
MESS04   DC    C'********* PRODUCT QUANTITY IS NOT A VALID NUMBER'      IVC12420
MESS05   DC    C'********* PRODUCT QUANTITY NOT VALID WITH DELETE'      IV012430
MESS06   DC    C'********* PRODUCT QUANTITY FIELD IS INCORRECT'         IVC12440
MESS07   DC    C'********* PRODUCT NAME NOT SPECIFIED FOR ADD'          IV012450
MESS08   DC    C'********* PRODUCT QUANTITY NOT VALID WITH ADD'         IV012460
MESS09   DC    C'********* PRODUCT NAME NOT SPECIFIED FOR NAME CHANGE'
MESS10   DC    C'********* PRODUCT QUANTITY NOT VALID WITH NAME CHANGE'
MESS11   DC    C'********* PRODUCT NAME NOT VALID  FOR INV INC/DEC'
MESS14   DC    C'********* PRODUCT QUANTITY ZERO FOR INV INC/DEC'
         LTORG                                                          IV012470
         EJECT                                                          IV012480
         PRINT NOGEN                                                    IV012490
INPUT    DCB   DSORG=PS,MACRF=GM,EODAD=EOJ,LRECL=80,DDNAME=SYSIN        IVC12500
ERROR    DCB   DSORG=PS,MACRF=PM,LRECL=121,RECFM=FBM,DDNAME=ERRPRNT     IV012520
OUTPUT   DCB   DSORG=PS,MACRF=(WP),BLKSIZE=100,RECFM=U,DDNAME=OP        IV012540
         PRINT GEN                                                      IV012560
         EJECT                                                          IV012570
         ENTRY IV01EP                                                   IV012580
IV01EP   DS    0H                     INITIALIZATION                    IV012590
         SAVE  (14,12)                                                  IV012600
         DROP  11                                                       IV012610
         BALR  2,0                                                      IV012620
         USING *,2                                                      IV012630
         ST    13,SYSREG13                                              IV012640
         L     15,=A(SAVEAREA)                                          IVC12650
         L     11,=A(IV01PJB)         LOAD COVER REGISTER               IV012660
         DROP  2
         USING IV01PJB,11
         OPEN  (INPUT,(INPUT),OUTPUT,(OUTPUT),ERROR,(CUTPUT))           IV012680
         USING IHADCB,3
         L     3,=A(CUTPUT)
         TM    DCBOFLGS,X'10'
         BO    INIT01
         WTO   'UNSUCCESSFULL CUTPUT OPEN'
         ABEND 4,DUMP
INIT01   L     3,=A(INPUT)
         TM    DCBOFLGS,X'10'
         BC    INIT02
         WTO   'UNSUCCESSFULL INPUT CPEN'
         ABEND 8,DUMP
INIT02   L     3,=A(ERROR)
         TM    DCBOFLGS,X'10'
         BC    MAIN
         WTO   'UNSUCCESSFULL ERROR PRINT OPEN'
         ABEND 12,DUMP
EOJ      DS    0H
* INPUT END CF FILE RCUTINE
         CLOSE INPUT
         CLCSE OUTPUT
         CLOSE ERROR
         L     13,SYSREG13
         RETURN (14,12)
         SPACE 6
SYSREG13 DS    1F                                                       IV012710
SAVEAREA DS    18F                                                      IV012720
         SPACE 3                                                        IV012730
         EJECT
         DCBD  DSORG=PS
         END   IV01EP                                                   IV012740
```

PG.BX	NAME	PG.BX	NAME	PG.BX	NAME	PG.BX	NAME	PG.BX	NAME
1.16	ACEHRCH	2.16	DEL10	2.01	IINC30	4.01	IV01EP	3.13	WRITE
1.15	ADD	2.23	DEL15	2.06	IINC50	1.08	IV01PJB	3.22	WRITE01
1.29	ADD05	2.20	DEL20	2.11	IINC60	1.09	MAIN	3.25	WRITE02
1.28	ADD10	2.22	DEL30	1.18	INERROR	3.01	NAMCHG	3.17	WRITE10
1.21	ADD15	4.17	EOJ	4.09	INIT01	3.05	NAM05	5.03	WRITE11
1.25	ADD30	1.04	IINC05	4.13	INIT02	3.06	NAM10	3.33	WRITE20
5.05	BLANK	1.06	IINC07	5.01	INVDEC	3.10	NAM30	3.30	WRITE33
2.13	DEL	1.07	IINC10	1.01	INVINC	5.04	SPACE	5.02	WRITE44
2.15	DEL05	2.09	IINC20						

```
B-62 CARD ID   PAGE/BOX   NAME          REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)

ASSEMBLY MODULE   ASSEMBLY PROGRAM

CHART TITLE - 'VERIFY INVENTORY SYSTEM UPDATES'

(000085)   1.01  INVINC     IV010890  1.14     (000117)  5.01
(000090)   1.04  IINC05     (000086)  1.01
(000095)   1.06  IINC07
(000096)   1.07  IINC10
IV010730   1.08  IV01PJB
IV010760   1.09  MAIN       IV010980  1.18     (000217)  3.25     IV011740  3.35     (000289)  4.14
IV010950   1.16  ACERROR    IV010860  1.13
IV010970   1.19  INERROR    IV011720  3.33
IV011770   1.19  ADD        IV010860  1.13
IV011830   1.21  ADD15      (000154)  1.29
IV011930   1.25  ADD30
IV011810   1.28  ADD10      IV011780  1.20
IV011820   1.29  ADD05
(000112)   2.01  IINC30     (000099)  2.03
(000104)   2.06  IINC50     (000114)  2.02     (000111)  2.10     (000116)  2.11
(000109)   2.09  IINC20     (000098)  2.03
(000115)   2.11  IINC60     (000101)  2.04
(000108)   2.12             (000105)  2.07
IV011020   2.13  DEL        IV010920  1.15
IV011060   2.15  DEL05
IV011080   2.16  DEL10      IV011030  2.13
IV011180   2.20  DEL20
IV011130   2.22  DEL30
IV011190   2.23  DEL15      IV011100  2.18     (000131)  2.21
IV011970   3.01  NAMCHG     IV010890  1.14
IV012020   3.05  NAM05
IV012030   3.06  NAM10      IV012000  3.03
IV012130   3.10  NAM30
IV011230   3.13  WRITE      IV011850  1.23     (000148)  1.26     IV011940  1.27     (000107)  2.08     IV011200  2.23
                            (000165)  3.08     (000169)  3.11     IV012140  3.12
IV011380   3.17  WRITE10    IV011420  3.19
IV011680   3.22  WRITE01    IV011330  3.14     IV011350  3.15     IV011380  3.17     IV011400  3.18     IV011480  3.26
                            IV011520  3.28     IV011550  3.30     (000205)  3.31     (000207)  3.32     IV011220  5.03
IV011710   3.25  WRITE02    IV011680  3.22
IV011520   3.28             IV011500  3.27
IV011550   3.30  WRITE33    (000200)  3.27
IV011720   3.33  WRITE20
IV012600   4.01  IV01EP
(000283)   4.09  INIT01     (000279)  4.06
(000289)   4.13  INIT02     (000284)  4.10
(000294)   4.17  EOJ

CHART TITLE - EQU STATEMENTS

(000117)   5.01  INVDEC     IV010920  1.15
(000218)   5.02  WRITE44    IV011440  3.20
IV011220   5.03  WRITE11    IV011460  3.21
(000237)   5.04  SPACE
(000238)   5.05  BLANK

CHART TITLE - CONSTANTS AND STORAGE AREAS

ASSEMBLY MODULE   ASSEMBLY SUBROUTINE

CHART TITLE - 'CONVERT ALPHA FORMAT NUMBER TO DECIMAL'

(000002)   7.01  DECCVT     IV011840  1.22-X   IV011090  2.17-X   IV012040  3.07-X
(000010)   7.05  CHK00      (000022)  7.13
(000016)   7.08  CHK02
(000016)   7.09             (000012)  7.06
(000018)   7.10  CHK01      (000010)  7.05
(000020)   7.11  CHK03      (000014)  7.07
(000021)   7.12  CHK04      (000019)  7.10
(000032)   7.18  EXIT       (000030)  7.16
```

LOCATION CARD ID	PAGE/BOX	DIAGNOSTIC
(0000471)	1.07	UNDEFINED - 'REG. VALUE'
(0002181)	5.02	UNDEFINED - 'WRITE26'

CHART TITLE - 'VERIFY INVENTORY SYSTEM UPDATES'

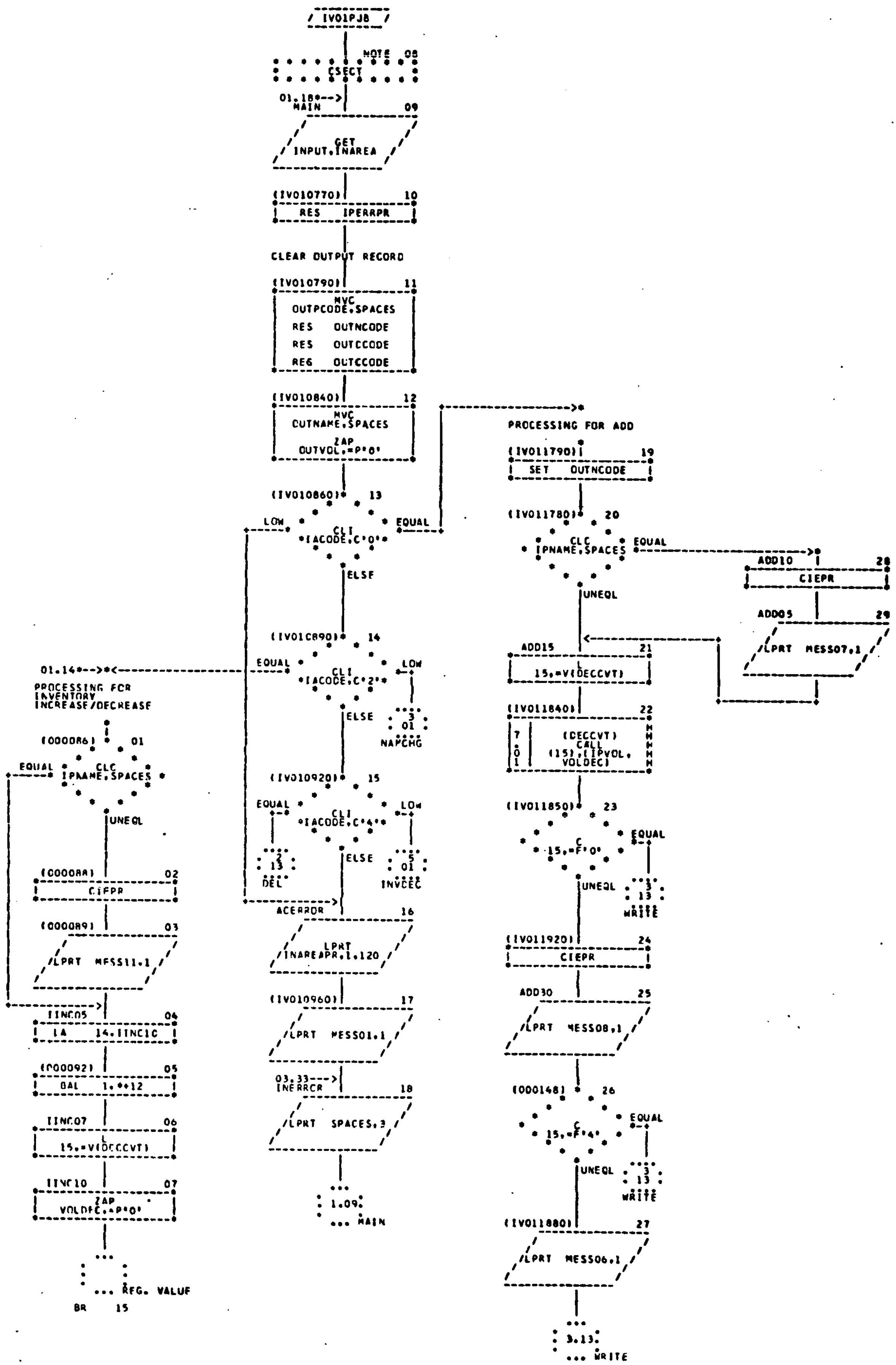
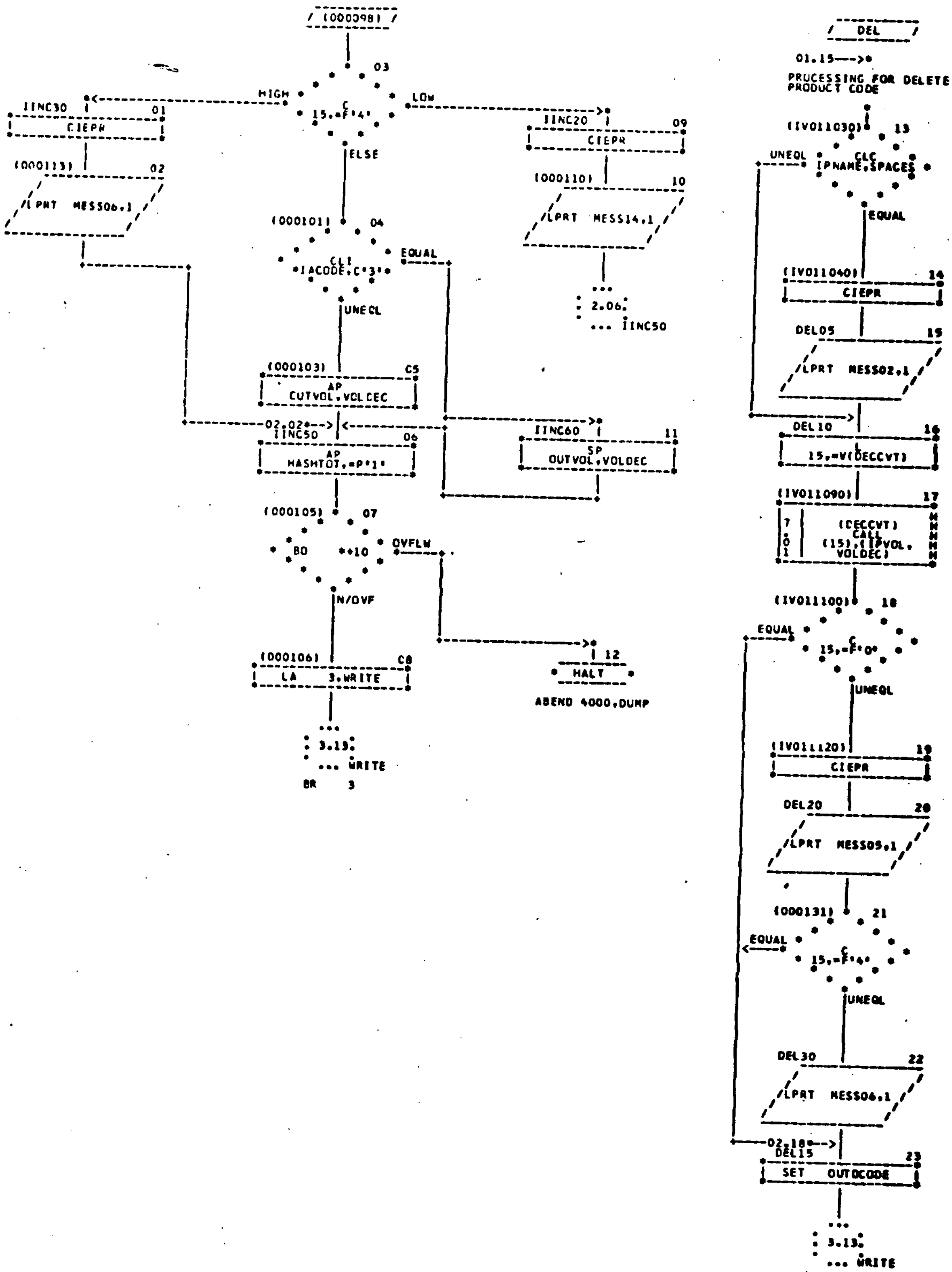


CHART TITLE - 'VERIFY INVENTORY SYSTEM UPDATES'



```
09/08/70        MACRO USAGE SUMMARY                       AUTOFLOW CHART SET - SAMPLE   ASSEMBLY PROGRAM              PAGE   1
CARD ID         MACRO          INVOCATIONS (SOURCE SEQUENCE NO.)
(LIBRARY)       ABEND          (000108)    (000282)    (000287)    (000292)
(LIBRARY)       CALL           IV011090    IV011840    IV012040
(LIBRARY)       CHECK          IV010735
(LIBRARY)       CIEPR          (000088)    (000109)    (000112)    IV010650    IV010650    IV011040
                               IV011120    IV011810    IV011920    IV012010    IV012120
(LIBRARY)       CLOSE          (000295)    (000296)    (000297)
(LIBRARY)       DCB            IV012500    IV012520    IV012540
(LIBRARY)       GET            IV010760
IV010590        IFERR          IV011680    IV011720
(LIBRARY)       LPRT           (000089)    (000110)    (000113)    IV010180    IV010180    IV010680
                               IV010950    IV010960    IV010970    IV011060    IV011130    IV011180
                               IV011690    IV011710    IV011820    IV011880    IV011930    IV012020
                               IV012080    IV012130
(LIBRARY)       OPEN           IV012680
(LIBRARY)       PUT            IV010450
IV010490        RES            IV010770    IV010800    IV010810    IV010820
(LIBRARY)       RETURN         (000299)
(LIBRARY)       SAVE           IV012600
IV010540        SET            IV010690    IV011190    IV011700    IV011790    IV011990
(LIBRARY)       WRITE          IV010730
(LIBRARY)       WTO            (000281)    (000286)    (000291)
```

```
09/08/70        LITERAL SUMMARY                           AUTOFLOW CHART SET - SAMPLE   ASSEMBLY PROGRAM              PAGE   1
                  LITERAL                                   REFERENCES  (SOURCE SEQUENCE NO.)
=A(ERROR)                                          (000288)
=A(INPUT)                                          (000283)
=A(IPCODE+1,1,IPCODE+3)                            IV011370
=A(IV01PJB)                                        IV012660
=A(OUTPUT)                                         (000278)
=A(SAVEAREA)                                       IV010440    IV012650
=F'0'                                              (000165)    IV011100    IV011850
=F'4'                                              (000098)    (000131)    (000148)    (000169)
=P'0'                                              (000096)    IV010850
=P'1'                                              (000104)
=V(DECCVT)                                         (000095)    IV011080    IV011830    IV012030
```

```
09/08/70        MODIFIED TAG SUMMARY                      AUTOFLOW CHART SET - SAMPLE   ASSEMBLY PROGRAM              PAGE   1
TAG                 OPERAND                                 REFERENCES  (SOURCE SEQUENCE NO.)
INAREA    INAREA+80                                IV012220
IPCODE    IPCODE+1                                 IV011370
          IPCODE+3                                 IV011370
          IPCODE+4                                 IV011440    IV011460
          IPCODE+5                                 IV011480    IV011500
          IPCODE+6                                 IV011520    IV011540
          IPCODE+7                                 (000205)    (000207)
OUTAREA   OUTAREA+100                              IV012310
PRBUF     PRBUF+1                                  IV010210    IV010220
          PRBUF+1(&FUDGE&RECORD)                   IV010270
          PRBUF+1(&LENGTH)                         IV010240
          PRBUF+2(&FUDGE.PRBUF-2)                  IV010220
```

```
09/10/70        INPUT LISTING                         AUTOFLOW CHART SET - SAMPLE   ASSEMBLY WITH CHART CODES

CHART/ASSEMBLY -    (LIST1)

 CARD NO      ****                            CONTENTS                                       ****

     1        CVT      TITLE 'CONVERT ALPHA FORMAT NUMBER TO DECIMAL'              RV       0010
     2        *                                     (DECCVT)                       EN       0020
     3        DECCVT   CSECT ,                      ROUTINE ENTRANCE POINT         T        0030
     4                 SAVE  (14,12)                                                        0040
     5                 USING DECCVT,11                                                      0050
     6                 ST    1,SYSREG1                                                      0060
     7                 L     2,0(1)                                                         0070
     8                 MVC   INVALUE,0(2)                                                   0080
     9                 LM    3,5,=A(INVALUE,1,INVALUE+L'INVALUE-1)                          0090
    10                 LA    6,INVALC                                                       0100
    11        CHK00    CLI   0(3),X'40'             VALIDITY CHECK ALL DIGITS      P        0110
    12                 BE    CHK01                  AND LEADING ZERO FILL FIELD    C        0120
    13                 CLI   0(3),X'F0'                                                     0130
    14                 BL    CHK02                                                          0140
    15                 CLI   0(3),X'F9'                                                     0150
    16                 BNH   CHK03                  ALL DIGITS VALID DECIMAL       CN       0160
    17        CHK02    L     15,=F'8'               SET RC = 8                     P        0170
    18                 RETURN (14,12),RC=(15)                                      EV       0180
    19        CHK01    MVI   0(6),C'0'                                                      0190
    20                 B     CHK04                                                          0200
    21        CHK03    MVC   0(1,6),0(3)                                           N        0210
    22        CHK04    LA    6,1(6),0(3)                                                    0220
    23                 BXLE  3,4,CHK00                                                      0230
    24        * DATA HAS NOW BEEN VALIDITY CHECKED                                 C        0240
    25                 PACK  OUTVALD,INVALC         CONVERT VALUE TO DECIMAL       P        0250
    26                 L     1,SYSREG1                                                      0260
    27                 L     2,4(1)                                                         0270
    28                 MVC   0(L'OUTVALD,2),OUTVALD                                         0280
    29                 SR    15,15                  SET RC = 0                     P        0290
    30                 AP    OUTVALD,=P'0'          IS VALUE ZERO                  CN       0300
    31                 BZ    EXIT                                                  C        0310
    32                 LA    15,4                   SET RC = 4                     P        0320
    33        EXIT     RETURN (14,12),RC=(15)                                      EV       0330
    34                 SPACE 6                                                              0340
    35        INVALUE  DS    CL8                                                            0350
    36        INVALC   DS    CL8                                                            0360
    37        OUTVALD  DC    P'0000000000'                                                  0370
    38                 SPACE 3                                                              0380
    39                 END                                                                  0390
```

B-68

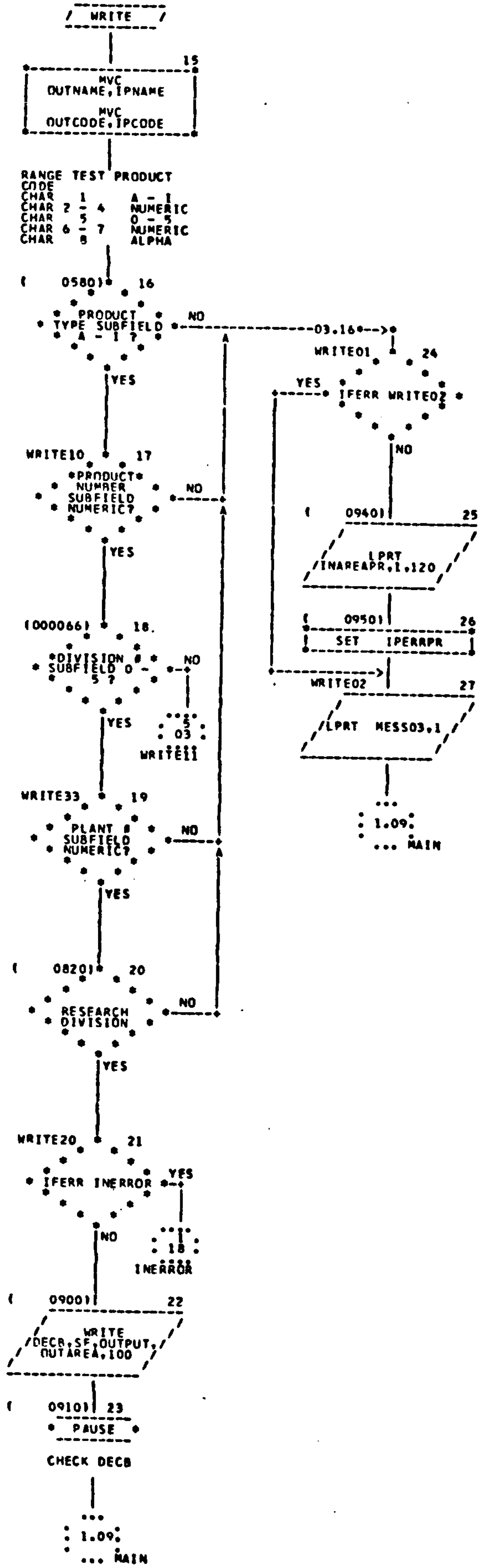
ART TITLE - TITLE 'CONVERT ALPHA FORMAT NUMBER TO DECIMAL'

```
            -----------
           /  DECCVT  /
            -----------
                 *
   ROUTINE ENTRANCE
   POINT
                 *
      CHK00      I          01
   *-------------*-------------*
   I    VALIDITY CHECK         I
   I    ALL DIGITS AND         I
   I   LEADING ZERO FILL       I
   I        FIELD              I
   *---------------------------*
                 I
                 I
                 *    02
              *     *
           *           *
        *ALL DIGITS *  YES
      * VALID DECIMAL  *---------------------------->*
           *           *                  CHK03      I    NOTE  05
              *     *                  * * * * * * * * * * * * *
                 *                     * DATA HAS NOW BEEN     *
                 INO                   * VALIDITY CHECKED      *
                 I                     * * * * * * * * * * * * *
                 I                                   I
                 I                                   I          06
      CHK02      I          03         *-----------------------*
   *-------------*-------------*       I CONVERT VALUE TO      I
   I      SET RC = 8           I       I     DECIMAL           I
   *---------------------------*       *-----------------------*
                 I                                   I
                 I  04                               I          07
            -----------                *-----------------------*
           *   EXIT    *               I     SET RC = 0        I
            -----------                *-----------------------*
                                                     I
             RETURN                                  I
         (14,12),RC=(15)                             *    08
                                                  *     *
                                               *           *
                                    YES      *               *
                                  +------* IS VALUE ZERO *
                                  I          *               *
                                  I            *           *
                                  I               *     *
                                  I                  *
                                  I                  INO
                                  I                  I
                                  I                  I
                                  I                  I          09
                                  I    *-----------------------*
                                  I    I     SET RC = 4        I
                                  I    *-----------------------*
                                  I                  I
                                  +----------------->I
                                              EXIT   I  10
                                               -----------
                                              *   EXIT    *
                                               -----------

                                                RETURN
                                            (14,12),RC=(15)
```

```
CARD NO     ****                         CONTENTS                                        ****

            *
            WRITE    MVC   OUTNAME,IPNAME                                        M     0440
                     MVC   OUTCODE,IPCODE                                              0450
            * RANGE TEST PRODUCT CODE                                                  0460
            * CHAR  1      A - I                                                       0470
            * CHAR 2 - 4   NUMERIC                                                     0480
            * CHAR  5      0 - 5                                                       0490
            * CHAR 6 - 7   NUMERIC                                                     0500
            * CHAR  8      ALPHA                                                       0510
            *                                                                          0520
            *                                                                          0530
            *                                                                    *AE
            *                                                                    *C
                     CLI   IPCODE,X'C1'                                                0560
                     BC    4,WRITE01                                                   0570
            *                                        PRODUCT TYPE SUBFIELD       DY    0580
                     CLI   IPCODE,X'C9'              A - I ?                     C     0590
                     BC    2,WRITE01                                                   0600
                     LM    3,5,=A(IPCODE+1,1,IPCODE+3)                                 0610
            WRITE10  CLI   0(3),X'F0'                                                  0620
                     BL    WRITE01                                                     0630
                     CLI   0(3),X'F9'                                                  0640
                     BH    WRITE01                   PRODUCT NUMBER SUBFIELD     DY    0650
                     BXLE  3,4,WRITE10               NUMERIC?                    C     0660
                     CLI   IPCODE+4,C'0'                                               0670
                     BL    WRITE44
                     CLI   IPCODE+4,C'5'             DIVISION # SUBFIELD         DY    0690
                     BH    WRITE11                   0 - 5 ?                     C
                     CLI   IPCODE+5,X'F0'                                              0710
                     BL    WRITE01                                                     0720
                     CLI   IPCODE+5,X'F9'                                              0730
                     BNH   *+8                                                         0740
                     B     WRITE33                                                     0750
                     CLI   IPCODE+6,X'F0'                                              0760
                     BL    WRITE01                                                     0770
                     CLI   IPCODE+6,X'F9'                                              0780
            WRITE33  BH    WRITE01                   PLANT # SUBFIELD NUMERIC?   DY    0790
                     TM    IPCODE+7,X'C0'                                              0800
                     BNO   WRITE01                   RESEARCH DIVISION           DY    0810
                     CLI   IPCODE+7,X'F0'                                              0820
                     BL    WRITE01                   SUBFIELD ALPHABETIC?        C     0830
            *                                                                          0840
            *                                                                          0850
            *                                                                    *A
            *                                                                    *CE
            WRITE20  IFERR INERROR                                                     0880
                     WRITE DECB,SF,OUTPUT,OUTAREA,100                                  0890
                     CHECK DECB                                                        0900
                     B     MAIN                                                        0910
            WRITE01  IFERR WRITE02
                     LPRT  INAREAPR,1,120                                              0930
                     SET   IPERRPR                                                     0940
            WRITE02  LPRT  MESS03,1                                                    0950
                     B     MAIN                                                        0960
            WRITE44  EQU   WRITE26                                                     0970
            WRITE11  EQU   WRITE01
                     EJECT
            *
                                 *AE
```

CHART TITLE - 'VERIFY INVENTORY SYSTEM UPDATES'

B-70



CARD NO **** CONTENTS ****

```
OPTION PRINTDCL,TEXT=(1,80)
MERGE: PROCEDURE;
                          /* FILE DECLARATIONS*/
       DECLARE MAINFILE FILE INPUT BUFFERED ENVIRONMENT
             (MEDIUM(SYS003,2311)F(78));
       DECLARE INSFILE  FILE INPUT BUFFERED SEQUENTIAL ENVIRONMENT
             (MEDIUM(SYS003,2311)F(112));
       DECLARE CHGFILE  FILE INPUT BUFFERED SEQUENTIAL ENVIRONMENT
             (MEDIUM(SYS003,2311)F(112));
       DECLARE MRGFILE  FILE OUTPUT BUFFERED SEQUENTIAL ENVIRONMENT
             (MEDIUM(SYS021,2400)F(1690,84));
       DECLARE PRINTER FILE OUTPUT PRINT STREAM ENVIRONMENT
             (MEDIUM(SYSLST,1403)F(130));
                          /* EXTERNAL DECLARATIONS*/
       DECLARE
             ((DEBUG,SKIP) BIT(8),REGNUM FIXED DECIMAL(2,0)) EXTERNAL;
                          /* LOCAL DECLARATIONS*/
       DECLARE
             IFPRNT BIT(8),
             (REGCNT,ONNUM,MAPNUM,DCLNUM,FORMNUM,RCOUNTM,RNUM)
             FIXED DECIMAL(2,0);
                          /*STRUCTURE DECLARATIONS*/
       DECLARE
           1 MAIN,
             2 SIZ          FIXED DECIMAL(2,0),
             2 ONF          FIXED DECIMAL(1,0),
             2 SRT          FIXED BINARY(8,0),
             2 CNT          FIXED DECIMAL(2,0),
             2 NUM          FIXED DECIMAL(2,0),
             2 AFC          CHAR(2),
             2 DST,
               3 DST1       CHAR(1),
               3 DST9       CHAR(9),
             2 TXT          CHAR(10);
       DECLARE
           1 INS,
             2 SIZ          FIXED DECIMAL(2,0),
             2 ONF          FIXED DECIMAL(1,0),
             2 SRT          FIXED BINARY(8,0) INITIAL('00000000'B),
             2 CNT          FIXED DECIMAL(2,0),
             2 NUM          FIXED DECIMAL(2,0),
             2 AFC          CHAR(2),
             2 DST          CHAR(10),
             2 TXT          CHAR(10);
       DECLARE
           1 CHG,
             2 SIZ          FIXED DECIMAL(2,0),
             2 NUM          FIXED DECIMAL(2,0),
             2 TYP          FIXED BINARY(8,0) INITIAL('11111111'B),
             2 FILL         CHAR(5),
             2 DST,
               3 DLA        CHAR(2),
               3 DUMMY      CHAR(8);
       DECLARE
             MRGBUF         CHAR(50);
                            /* LABEL VARIABLES*/
       DECLARE
             (LABMAIN,LABINS,LABCHG) LABEL;

       ON ENDFILE(MAINFILE)
             BEGIN;
               CLOSE FILE(MAINFILE);
               CLOSE FILE(INSFILE);
               CLOSE FILE(CHGFILE);
               CLOSE FILE(MRGFILE);
               DISPLAY ('END OF MERGE');
                            /* EXTERNAL PROCEDURES */
               CALL SORT;
               CALL FORMAT;
               CALL RELEASE;
               GO TO EXIT;
             END;

       ON ENDFILE(INSFILE)
             BEGIN;
               IF INS.SRT = '00000000'B THEN
                 DO;
                   DISPLAY ('NO INSERTIONS');
                   DEBUG = DEBUG | '00010000'B;
                 END;
               INS.SIZ = 0;
               INS.ONF = 0;
               /* SET END OF FILE FLAG*/
               INS.SRT = '11111111'B;
               INS.CNT = 0;
               INS.NUM = 0;
               INS.AFC = '  ';
               INS.DST = '          ';
             END;

       ON ENDFILE(CHGFILE)
             BEGIN;
               IF CHG.TYP = '11111111'B THEN
                 DO;
                   DISPLAY ('NO CHANGES');
                   DEBUG = DEBUG | '00100000'B;
                 END;
               IF (DEBUG & '00010000'B) ¬= 0 &
                  (IFPRNT & '00100000'B) ¬= 0 THEN
                   GO TO EXIT;
               ELSE
                 DO;
                   CHG.SIZ = 0;
                   CHG.NUM = 0;
                   /* RETAIN CHG.TYP = '11111111'B */
                   CHG.CNT = 0;
                   CHG.DST = '          ';
                 END;
             END;

       IFPRNT = DEBUG;
       ONNUM = REGNUM+1;
       MAPNUM = ONNUM+1;
       DCLNUM = MAPNUM+1;
       FORMNUM,RCOUNTM = DCLNUM+1;
       LABMAIN = GETINS;
       LABINS = GETCHG;
       LABCHG = MERGO;

       OPEN FILE(MAINFILE), FILE(INSFILE), FILE(CHGFILE), FILE(MRGFILE);
GETMAIN: READ FILE(MAINFILE) INTO MAIN;
       GO TO LABMAIN;
GETINS:  IF INS.SRT ¬= '11111111'B THEN
           READ FILE(INSFILE) INTO INS;
       GO TO LABINS;
GETCHG:  IF CHG.TYP ¬= '11111111'B THEN
           READ FILE(CHGFILE) INTO CHG;
       GO TO LABCHG;

MERGO:   IF MAIN.SRT ¬= '01011001'B THEN GO TO MERG2;
           IF MAIN.AFC = 'R ' THEN
             PCNT = PCNT + 1 ;
           ELSE
             PCNT = MAIN.NUM + 1 ;
         RNUM = PCNT;
         CNT = MAIN.CNT;
MERG1:   IF INS.CNT ¬< MAIN.CNT THEN
           GO TO MERG1A;
         ELSE
           DO;
             CALL PUTINS;
             GO TO MERG1;
           END;
MERG1A:  CALL PUTMAIN;
         GO TO MERGO;
MERG2:   IF MAIN.AFC = '**' THEN GO TO MERG3;
           IF MAIN.AFC ¬= 'D ' THEN
             IF MAIN.AFC ¬= 'F ' THEN
               IF MAIN.ONF ¬= 1 THEN
                 DO;
                   RNUM = MAIN.NUM;
                   GO TO MERG1;
                 END;
```

B-72

```
                    ELSE
                       DO;
                          RNUM = ONNUM;
                          GO TO MERG1;
                       END;
                 ELSE
                    DO;
                       RNUM = FORMNUM;
                       GO TO MERG1;
                    END;
              ELSE
                 DO;
                    RNUM = OCLNUM;
                    GO TO MERG1;
                 END;
MERG3:     CALL PUTMAIN;
           DISPLAY('CANCEL MERGE');
EXIT:      RETURN;

PUTMAIN: PROCEDURE RECURSIVE;
           DECLARE LABPMAIN LABEL INITIAL (PMAINX1) ;
PMAIN0:    IF RNUM = 0 THEN
              DO;
                 DISPLAY ('MAIN RNUM = 0');
                 PUT FILE(PRINTER) EDIT ('MAIN.TXT=',MAIN.TXT)
                    (COLUMN(10),A(9),X(5),A(10));
                 PUT FILE(PRINTER) SKIP(2);
                 IFPRNT = IFPRNT | '00100000'B ;
              END;
PMAIN1:    MAIN.NUM = RNUM;
           IF MAIN.CNT = CHG.CNT THEN GO TO PMAIN3;
           IF MAIN.AFC = 'L ' THEN
              IF MAIN.TEXT ¬= '          ' THEN
                 MAIN.DST = MAIN.TXT;
              ELSE;
           ELSE;
           MRGBUF = MAIN;
PMAIN2:    WRITE FILE(MRGFILE) FROM (MRGBUF);
           GO TO LABPMAIN;
PMAIN3:    IF CHG.TYP ¬= '00000000'B THEN GO TO PMAIN4;
           IF MAIN.AFC = 'L ' &
              MAIN.DST1 ¬= '$' &
              MAIN.DST1 ¬= '.' &
              CHG.DLR = SUBSTR(MAIN.DST2,8,2) THEN
                DO;
                   MRGBUF = CHG;
                   LABCHG = PMAIN2;
                   GO TO GETCHG;
                END;
           ELSE
              GO TO PMAIN2;
PMAIN4:    CHG.NUM = RNUM;
           MRGBUF = CHG;
           WRITE FILE(MRGFILE) FROM (MRGBUF);
           LABCHG = PMAIN1;
           GO TO GETCHG;
PMAINX1: LABPMAIN = PMAINX2;
           DO WHILE INS.CNT ¬= STT;
              IF MAIN.CNT ¬= STT THEN GO TO PMAINX4;
              CALL PUTMAIN;
PMAINX2: END;
PMAINX3: CALL PUTINS;
PMAINX4: IF INS.CNT = STT THEN GO TO PMAINX3;
           LABPMAIN = PMAINX1;
           RETURN;
 END PUTMAIN;

PUTINS:  PROCEDURE;
PUTINS1: IF RNUM = 0 THEN
            DO;
               DISPLAY ('INSERTION RNUM = 0');
               PUT FILE(PRINTER) EDIT ('INS.TXT =',INS.TXT)
                  (COLUMN(10),A(9),X(5),A(10));
               PUT FILE(PRINTER) SKIP(2);
               IFPRNT = IFPRNT | '00100000'B;
            END;
         SAVE1 = INS.AFC;
         SAVE2 = INS.CNT;
         INS.NUM = RNUM;
         INS.ONE = MAIN.ONE;
         MRGBUF = INS;
         WRITE FILE (MRGFILE) FROM (MRGBUF);
         LABINS = PUTINS2;
         GO TO GETINS;
PUTINS2: IF INS.AFC = 'C ' THEN
            DO;
               INS.AFC = SAVE1;
               INS.CNT = SAVE2;
               /* COLLECT ALL 'C 'S AND OUTPUT */
               GO TO PUTINS1;
            END;
         ELSE
PUTINS3:    IF INS.AFC = 'S ' THEN
               DO;
                  LABINS = PUTINS3;
                  PUT FILE(PRINTER) EDIT ('INS.TXT =',INS.TXT)
                     (COLUMN(10),A(9),X(5),A(10));
                  PUT FILE(PRINTER SKIP(2);
                  /* SKIP ALL 'S 'S */
                  GO TO GETINS;
               END;
            ELSE
               RETURN;
 END PUTINS;

         END MERGE;
```

PG.BX	NAME	PG.BX	NAME	PG.BX	NAME
5.01	.ON-UNIT01	1.30	MERG1	2.06	PMAIN1
5.11	.ON-UNIT02	1.38	MERG1A	2.19	PMAIN2
5.18	.ON-UNIT03	1.23	MERG2	2.17	PMAIN3
1.34	EXIT	1.32	MERG3	2.20	PMAIN4
1.17	GETCHG	3.01	PMAINX1	4.01	PUTINS
1.14	GETINS	3.06	PMAINX2	4.01	PUTINS1
1.13	GETMAIN	3.07	PMAINX3	4.09	PUTINS2
1.01	MERGE	3.08	PMAINX4	4.12	PUTINS3
1.20	MERGO	2.01	PMAINO	2.01	PUTMAIN

B-74

LOCATION CARD ID	PAGE/BOX	DIAGNOSTIC
00000068	5.08	UNDEFINED - 'SORT' EXTERNAL REFERENCE
00000069	5.09	UNDEFINED - 'FORMAT' EXTERNAL REFERENCE
00000070	5.10	UNDEFINED - 'RELEASE' EXTERNAL REFERENCE

CARD ID	PAGE/BOX	NAME	REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)							

PL/I MODULE PL/I PROGRAM

CHART TITLE - PROCEDURE MERGE

00000002	1.01	MERGE								
00000116	1.06			7.03						
00000117	1.07			7.02						
00000118	1.08			7.01						
00000121	1.13	GETMAIN								
00000123	1.14	GETINS	00000237	4.09	00000253	4.14		7.03		
00000125	1.16		00000123	1.14						
00000126	1.17	GETCHG	00000201	2.18	00000209	2.22		7.02		
00000128	1.19		00000126	1.17						
00000130	1.20	MERGO	00000145	1.38		7.01				
00000146	1.23	MERG2	00000130	1.20						
00000134	1.29		00000131	1.21						
00000135	1.29		00000133	1.22						
00000137	1.30	MERG1	00000152 00000157	1.27 1.37	00000142	1.31	00000167	1.35	00000162	1.36
00000141	1.31	.10004		6.05						
00000169	1.32	MERG3	00000146	1.23						
00000169	1.32	.10006		6.04						
00000171	1.34	EXIT	00000071	5.10	00000100	5.24				
00000166	1.35		00000147	1.24						
00000161	1.36		00000148	1.25						
00000156	1.37		00000149	1.26						
00000144	1.38	MERG1A	00000137	1.30						
00000144	1.38	.10005		6.04						

CHART TITLE - PROCEDURE PUTMAIN RECURSIVE

00000173	2.01	PUTMAIN	00000169	1.32	00000144	1.38	00000213	3.05		6.04
00000175	2.01	PMAIN0								
00000183	2.06	PMAIN1		7.01						
00000183	2.06		00000175	2.01						
00000190	2.11		00000185	2.08	00000186	2.09				
00000193	2.12	PMAIN3	00000184	2.07						
00000200	2.18			7.01						
00000191	2.19	PMAIN2	00000204	2.23		7.01				
00000205	2.20	PMAIN4	00000193	2.12						
00000209	2.22			7.01						
00000204	2.23		00000194	2.13	00000194	2.14	00000195	2.15	00000196	2.16
00000210	3.01	PMAINX1		7.04						
00000210	3.01			7.04						
00000211	3.03		00000214	3.06						
00000213	3.05	.10007		6.04						
00000214	3.06	PMAINX2		7.04						
00000215	3.07	PMAINX3	00000216	3.08						
00000215	3.07		00000211	3.03						
00000215	3.07	.10008		6.05						
00000216	3.08	PMAINX4	00000212	3.04						
00000217	3.09			7.04						

CHART TITLE - PROCEDURE PUTINS

00000221	4.01	PUTINS	00000141	1.31	00000215	3.07		6.05
00000222	4.01	PUTINS1	00000243	4.11				
00000230	4.06		00000222	4.01				
00000236	4.08			7.02				
00000238	4.09	PUTINS2		7.02				
00000246	4.12	PUTINS3		7.02				
00000246	4.12		00000238	4.09				
00000248	4.13			7.02				
00000256	4.17		00000246	4.12				

CHART TITLE - ON-UNIT ACTION BLOCKS

00000061	5.01	.ON-UNIT01	00000040	1.01		
00000068	5.08	.10001		6.03		
00000069	5.09	.10002		6.01		
00000070	5.10	.10003		6.02		
00000075	5.11	.ON-UNIT02	00000074	1.02		
00000081	5.15		00000076	5.12		
00000092	5.18	.ON-UNIT03	00000091	1.03		
00000098	5.22		00000093	5.19		
00000103	5.25		00000099	5.22	00000098	5.23

CHART TITLE - CALLED PROCEDURES - CROSS REFERENCE

B-76

CHART TITLE - PROCEDURE MERGE

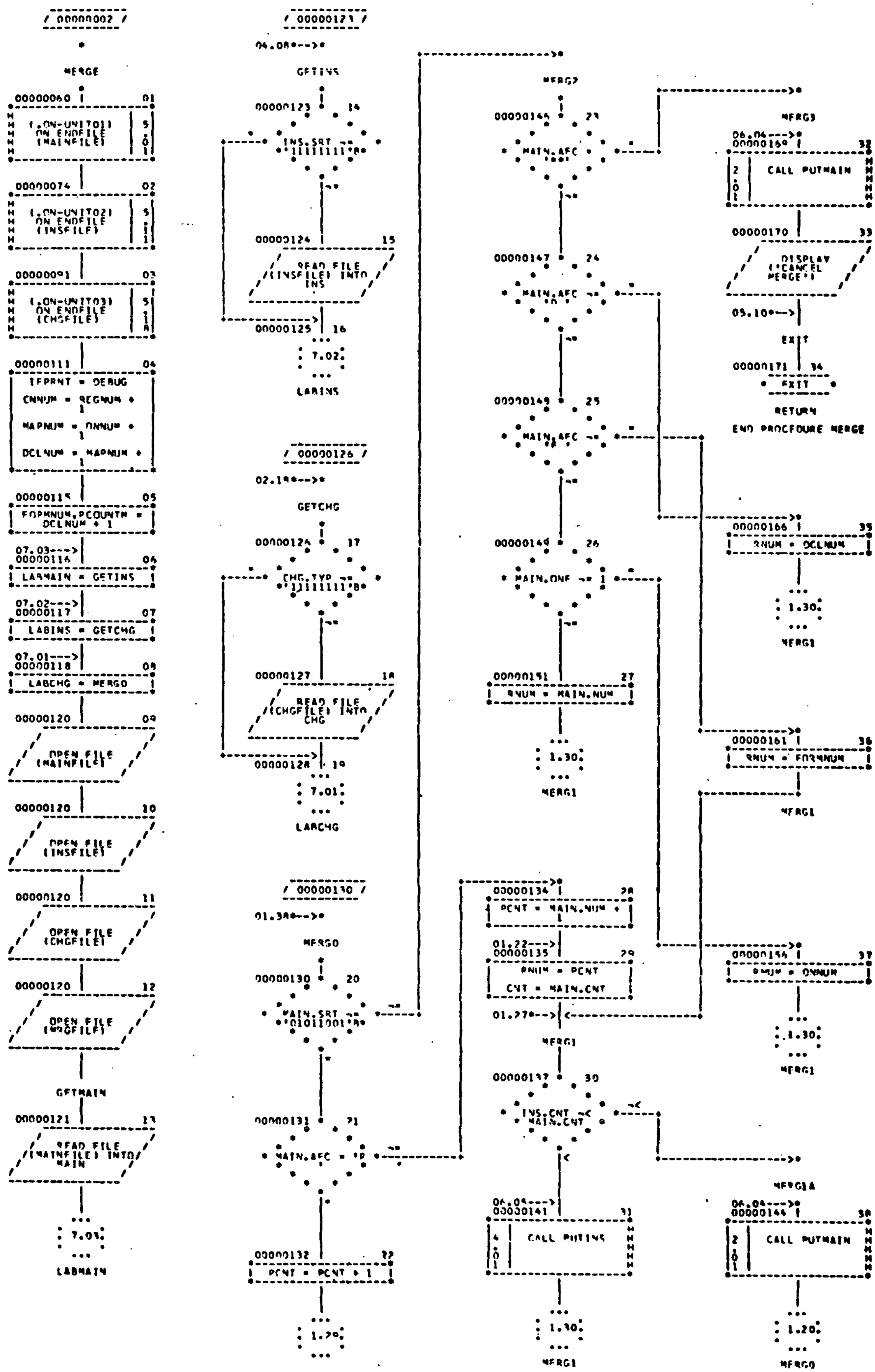


CHART TITLE - ON-UNIT ACTION BLOCKS

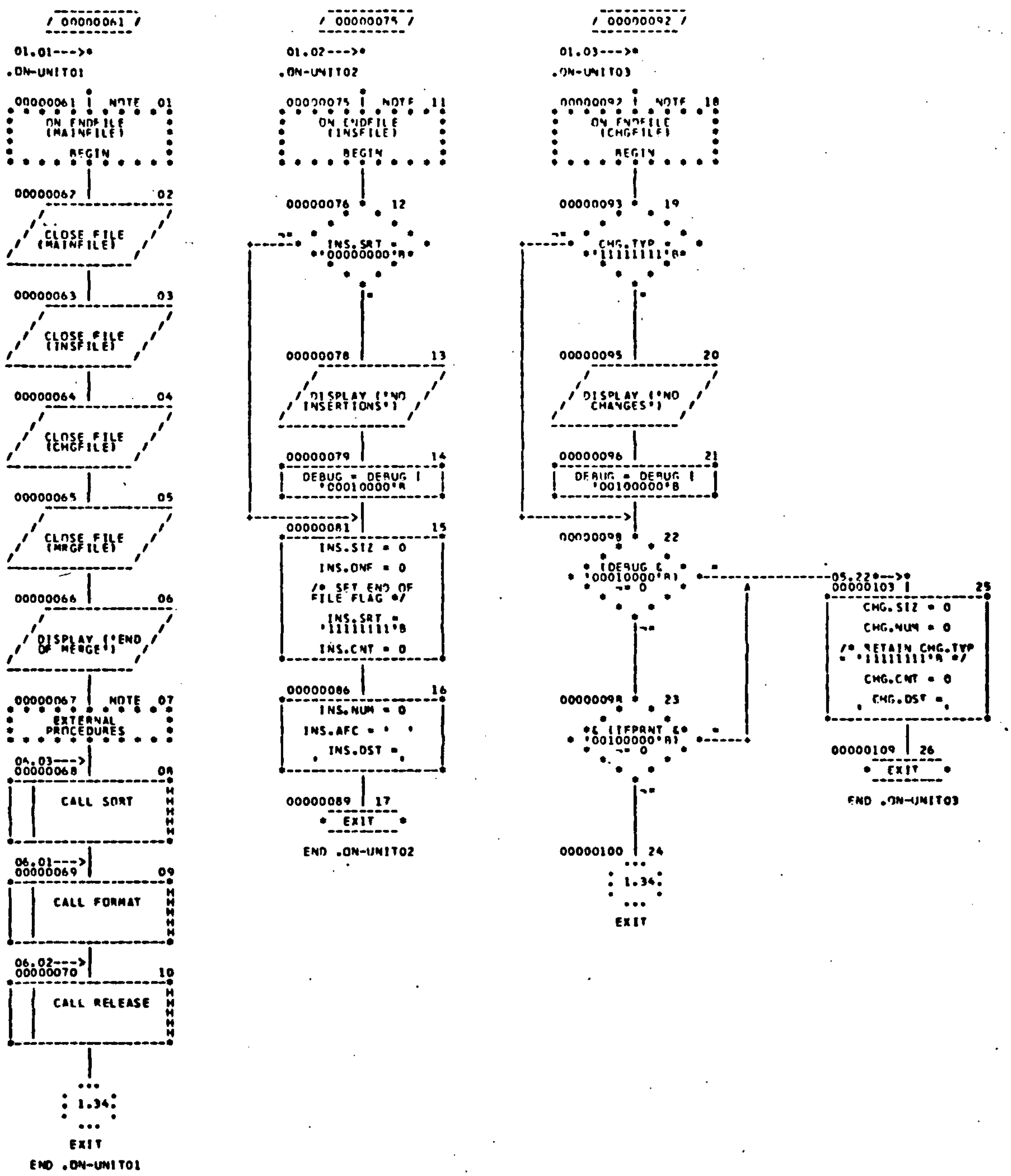
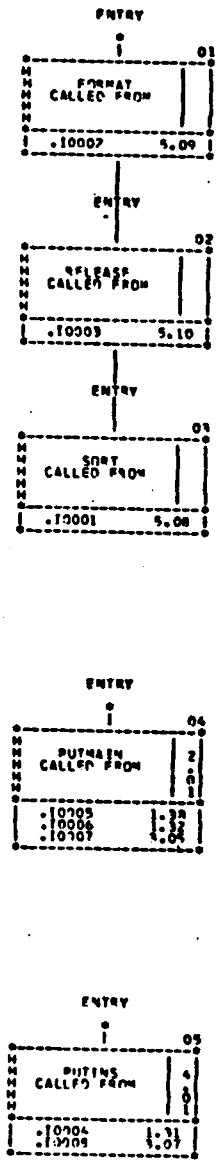


CHART TITLE - CALLED PROCEDURES - CROSS REFERENCE



```
                 -----------
                / LABCHG   /
                -----------
      01.19--->|
                                01
      *-------------------------*
      |          SCALAR         |
      |     LABEL VARIABLE      |
      *-------------------------*
      |  SET AT   TO GO TO      |
      |    1.08      1.20       |
      | 00000118 MERGO          |
      |    2.18      2.19       |
      | 00000200 PMAIN2         |
      |    2.22      2.06       |
      | 00000209 PMAIN1         |
      *-------------------------*


                 -----------
                / LABINS   /
                -----------
      01.16--->|
                                02
      *-------------------------*
      |          SCALAR         |
      |     LABEL VARIABLE      |
      *-------------------------*
      |  SET AT   TO GO TO      |
      |    1.07      1.17       |
      | 00000117 GETCHG         |
      |    4.08      4.09       |
      | 00000236 PUTINS2        |
      |    4.13      4.12       |
      | 00000248 PUTINS3        |
      *-------------------------*


                 -----------
                / LABMAIN  /
                -----------
      01.13--->|
                                03
      *-------------------------*
      |          SCALAR         |
      |     LABEL VARIABLE      |
      *-------------------------*
      |  SET AT   TO GO TO      |
      |    1.05      1.14       |
      | 00000116 GETINS         |
      *-------------------------*


                 -----------
                / LABPMAIN /
                -----------
      02.19--->|
                                04
      *-------------------------*
      |          SCALAR         |
      |     LABEL VARIABLE      |
      *-------------------------*
      |  SET AT   TO GO TO      |
      |    3.01      3.06       |
      | 00000210 PMAINX2        |
      |    3.09      3.01       |
      | 00000217 PMAINX1        |
      *-------------------------*
```

CHART TITLE - DUPLICATE DECLARATION MAP

B-80

```
                    PROCEDURE MERGE

00000023        SIZ OF MAIN
00000023        ONE OF MAIN
00000023        SRT OF MAIN
00000023        CNT OF MAIN
00000023        NUM OF MAIN
00000023        AFC OF MAIN
00000023        DST OF MAIN
00000023        TXT OF MAIN
00000035        SIZ OF INS
00000035        ONE OF INS
00000035        SRT OF INS
00000035        CNT OF INS
00000035        NUM OF INS
00000035        AFC OF INS
00000035        DST OF INS
00000035        TXT OF INS
00000045        SIZ OF CHG
00000045        NUM OF CHG
00000045        DST OF CHG
                PROCEDURE PUTMAIN


                END PROCEDURE
                PROCEDURE PUTINS


                END PROCEDURE

            END PROCEDURE
```

```
00000004   DECLARE MAINFILE FILE INPUT BUFFERED ENVIRONMENT (MEDIUM (SYS003,2311) F (78));
00000006   DECLARE INSFILE FILE INPUT BUFFERED SEQUENTIAL ENVIRONMENT (MEDIUM (SYS003,2311) F (112));
00000008   DECLARE CHGFILE FILE INPUT BUFFERED SEQUENTIAL ENVIRONMENT (MEDIUM (SYS003,2311) F (112));
00000010   DECLARE MRGFILE FILE OUTPUT BUFFERED SEQUENTIAL ENVIRONMENT (MEDIUM (SYS021,2400) F (1680,84));
00000012   DECLARE PRINTER FILE OUTPUT PRINT STREAM ENVIRONMENT (MEDIUM (SYSLST,1403) F (130));
00000015   DECLARE ((DEBUG,SKIP) BIT (8),REGNUM FIXED DECIMAL (2,0)) EXTERNAL;
00000018   DECLARE (ERRNT BIT (8),(REGCNT,ONNUM,MAPNUM,OCLNUM,FORMNUM,RCOUNTM,RNUM) FIXED DECIMAL (2,0));
00000023   DECLARE 1 MAIN,2 SIZ FIXED DECIMAL (2,0),2 ONE FIXED DECIMAL (1,0),2 SRT FIXED BINARY (9,0),
00000028             2 CNT FIXED DECIMAL (2,0),2 NUM FIXED DECIMAL (2,0),2 AFC CHAR (2),2 DST,3 DST1 CHAR (1),
00000033             3 DST2 CHAR (9),2 TXT CHAR (10);
00000035   DECLARE 1 INS,2 SIZ FIXED DECIMAL (2,0),2 ONE FIXED DECIMAL (1,0),2 SRT FIXED BINARY (8,
00000039             0) INITIAL ('00000000'B),2 CNT FIXED DECIMAL (2,0),2 NUM FIXED DECIMAL (2,0),2 AFC CHAR (2),
00000043             2 DST CHAR (10),2 TXT CHAR (10);
00000045   DECLARE 1 CHG,2 SIZ FIXED DECIMAL (2,0),2 NUM FIXED DECIMAL (2,0),2 TYP FIXED BINARY (8,
00000049             0) INITIAL ('11111111'B),2 FILL CHAR (5),2 DST,3 DLB CHAR (2),3 DUMMY CHAR (8);
00000054   DECLARE MRGBUF CHAR (50);
00000057   DECLARE (LABMAIN,LABINS,LABCHG) LABEL;
00000174   DECLARE LABPMAIN LABEL INITIAL (PMAINX1);
```

CHART TITLE - GET/PUT, FORMAT STATEMENTS

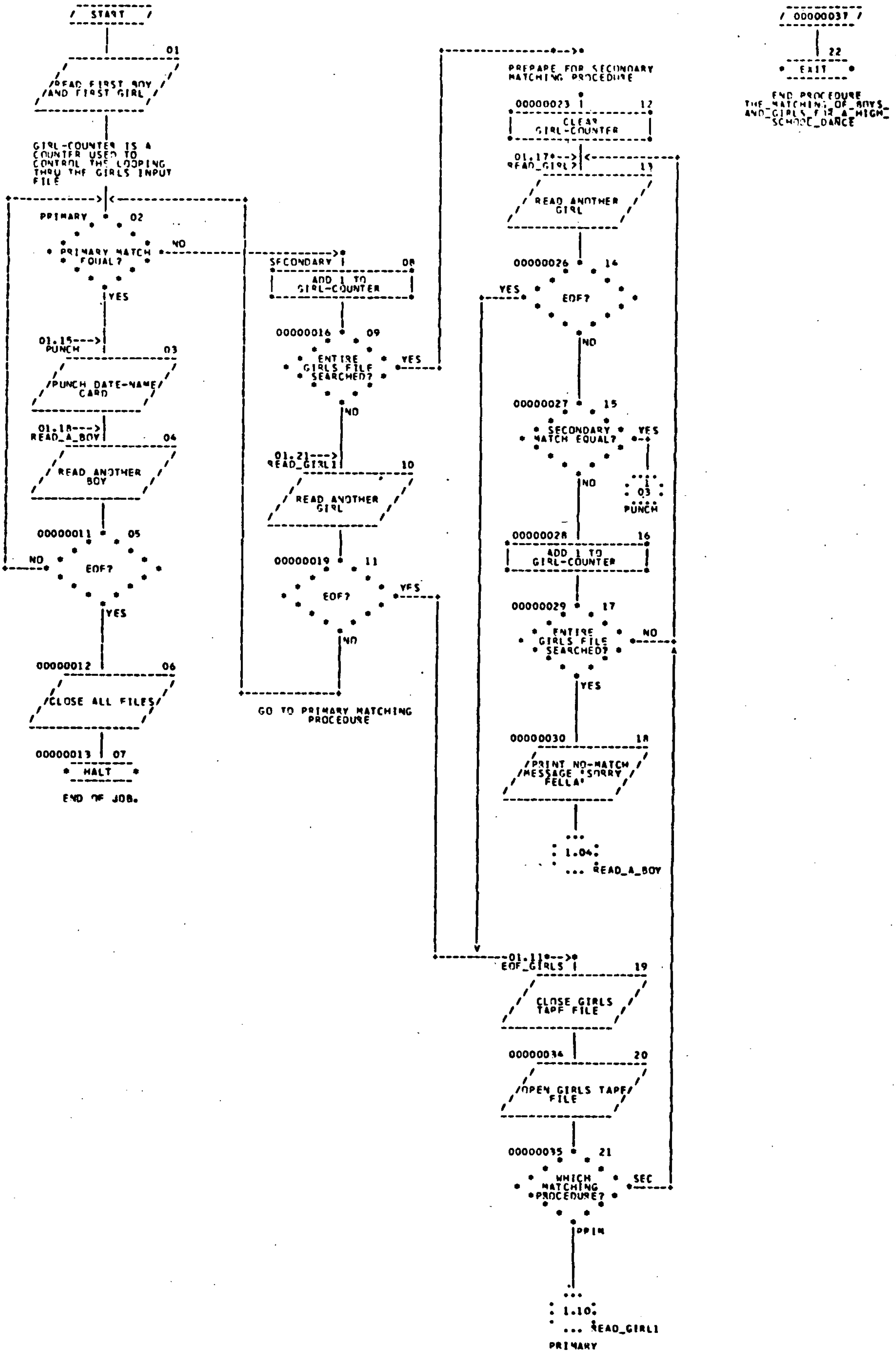
```
00000009 LIST       FORMAT (COLUMN (78),A,X (8),P 'ZZZ',X (5),P 'YYY',X
00000010 TOTAL      FORMAT (SKIP (2),COLUMN (23),A,COLUMN (41),P COLUMN (61)
00000178           .GET/PUT 001  PUT FILE (PRINTER) EDIT ('MAIN.TXT=',MAIN.TXT) (COLUMN (10),A (9),X (5),A (10))
00000180           .GET/PUT 002  PUT FILE (PRINTER) SKIP (2)
00000225           .GET/PUT 003  PUT FILE (PRINTER) EDIT ('INS.TXT =',INS.TXT) (COLUMN (10),A (9),X (5),A (10))
00000227           .GET/PUT 004  PUT FILE (PRINTER) SKIP (2)
00000249           .GET/PUT 005  PUT FILE (PRINTER) EDIT ('INS.TXT =',INS.TXT) (COLUMN (10),A (9),X (5),A (10))
00000251           .GET/PUT 006  PUT FILE (PRINTER SKIP (2)
```

CHART/PL1 MODULE (LIST2)

```
****                        CONTENTS                        ****
THE_MATCHING_OF_BOYS_AND_GIRLS_FOR_A_HIGH_SCHOOL_DANCE: PROCEDURE;
  START:
/*I  READ FIRST BOY AND FIRST GIRL                                   */
/*T  GIRL-COUNTER IS A COUNTER USED TO CONTROL THE LOOPING
     THRU THE GIRLS INPUT FILE                                       */
  PRIMARY:
/*DY  (SECONDARY) PRIMARY MATCH EQUAL?                               */
/*I  .PUNCH  PUNCH DATE-NAME CARD                                    */
  READ_A_BOY:
/*I  READ ANOTHER BOY                                                */
/*DY  (PRIMARY) EOF?                                                 */
/*I  CLOSE ALL FILES                                                 */
/*H  END OF JOB.                                                     */
  SECONDARY:
/*P  ADD 1 TO GIRL-COUNTER                                           */
/*DN  (PREPARE) ENTIRE GIRLS FILE SEARCHED?                          */
  READ_GIRL1:
/*I  READ ANOTHER GIRL                                               */
/*DN  (EOF_GIRLS) EOF?                                               */
/*B  (PRIMARY) GO TO PRIMARY MATCHING PROCEDURE                      */
  PREPARE:
/*T  PREPARE FOR SECONDARY MATCHING PROCEDURE                        */
/*P  CLEAR GIRL-COUNTER                                              */
  READ_GIRL2:
/*I  READ ANOTHER GIRL                                               */
/*DN  (EOF_GIRLS) EOF?                                               */
/*DN  (PUNCH)  SECONDARY MATCH EQUAL?                                */
/*P  ADD 1 TO GIRL-COUNTER                                           */
/*DY  (READ_GIRL2) ENTIRE GIRLS FILE SEARCHED?                       */
/*I  PRINT NO-MATCH MESSAGE 'SORRY FELLA'                            */
/*B  (READ_A_BOY)                                                    */
  EOF_GIRLS:
/*I  CLOSE GIRLS TAPE FILE                                           */
/*I  OPEN GIRLS TAPE FILE                                            */
/*D  (PRIM,SEC-READ_GIRL2) WHICH MATCHING PROCEDURE?                 */
/*B  (READ_GIRL1) PRIMARY                                            */
END THE_MATCHING_OF_BOYS_AND_GIRLS_FOR_A_HIGH_SCHOOL_DANCE;
```

CARD ID	PAGE/BOX	NAME	REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)			
		CHART/PL1 MODULE SAMPLE CHART PL/1 FACILITIES				
		CHART TITLE - PROCEDURE THE_MATCHING_OF_BOYS_AND_GIRLS_FOR_A_HIGH_SCHOOL_DANCE				
00000002	1.01	START				
00000006	1.02	PRIMARY	00000011	1.05	00000020	1.11
00000008	1.03	PUNCH	00000027	1.15		
00000009	1.04	READ_A_BOY	00000031	1.18		
00000014	1.08	SECONDARY	00000006	1.02		
00000017	1.10	READ_GIRL1	00000036	1.21		
00000024	1.13	READ_GIRL2	00000029	1.17	00000035	1.21
00000032	1.19	EOF_GIRLS	00000019	1.11	00000026	1.14

CHART TITLE - PROCEDURE THE_MATCHING_OF_BOYS_AND_GIRLS_FOR_A_HIGH_SCHOOL_DANCE



COMCHART

COMPRESS

GENERAL

COMCHART accepts source programs written in assembly language and COBOL. The package can also take a designer's specifications and produce flowcharts, using a special Design Language that involves a coding technique similar to that of most assembly languages. A listing of the source deck plus a cross-reference listing of all element and procedure names are also generated, but can be suppressed at the user's option.

The package runs on a IBM 360 (OS) with a minimum of 65K bytes of main core storage and 4 sequential files on any devices, IBM 360 (DOS) with a minimum of 65K bytes of core storage and 4 sequential work files (a 2311, a 2314, or 4 tapes), RCA Spectra 70 (TDOS) with a minimum of 65K bytes of main storage and 4 tapes or 1 disc.

The package consists of 3 flowcharter subsystems. The source language is a combination of assembly language and Cobol.

PACKAGE OUTPUT

Flowcharts - Detailer flowchart where each line of source code is represented by a symbol. Each flowchart bears a page number, a user-furnished identification of the program, and the name of the user's organizatio

Element Index - (COBOL only) an alphabetical cross-reference index of element names is given. Each index contains. each name in the program, its type (e.g. data level, paragraph name, file name), and the name of the procedure which references it.

Procedure Skeleton - This lists each procedure name in the procedure division, together with all procedures referenced by and refering to that procedure.

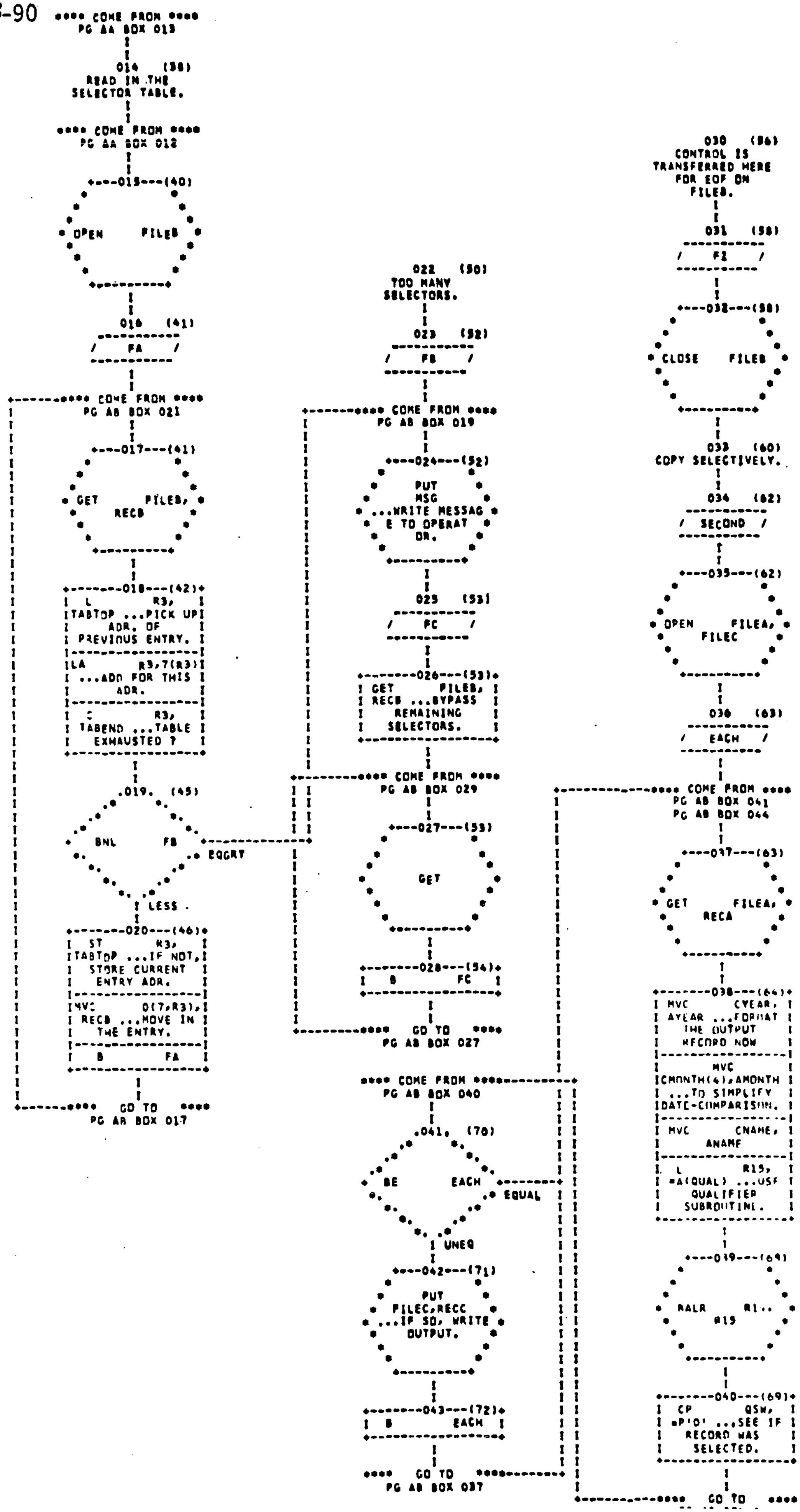
Source Deck Listing - An 80/80 list of the soruce deck is printed.

Diagnostics - This report follows the deck listing for a Design Language program, and given a narrative description of errors in the Design Language definition.

```
SAMPLE      TO DEMONSTRATE COMCHART FOR ASSEMBLY                         CHART/A

SAMPLE   START 0                                                          00010
R0       EQU   0                                                          00015
R1       EQU   1                                                          00020
R2       EQU   2                                                          00025
R3       EQU   3                                                          00030
R4       EQU   4                                                          00035
R5       EQU   5                                                          00040
R6       EQU   6                                                          00045
R7       EQU   7                                                          00050
R8       EQU   8                                                          00055
R9       EQU   9                                                          00060
R10      EQU   10                                                         00065
R11      EQU   11                                                         00070
R12      EQU   12                                                         00075
R13      EQU   13                                                         00080
R14      EQU   14                                                         00085
R15      EQU   15                                                         00090
*                                                                         00105
*              REGISTER USAGE                                             00110
*                R.0,1,2,13,14,15--AS DEFINED BY IBM.                     00115
*                R.10,11--OPEN ADDRESSING.                                00120
*                R.12--SELECTIVE ADDRESSING.                              00125
*                R.3,4,5,6,7,8,9--SPAN-OF-ATTENTION.                      00130
*                                                                         00135
FIRST    STM   R14,R12,12(R13)                                            00140
         BALR  R10,0            SET UP OPEN REGS.                         00143
         USING *,R10,R11                                                  00145
         LA    R3,4095                                                    00150
         LA    R11,1(R10,R3)                                              00155
         CNOP  2,4                                                        00160
         MVC   8(4,R13),=A(*+14)                                          00165
         ST    R13,*+12                                                   00168
         BAL   R13,FGO          SET UP SAVE AREA.                         00170
         DS    18F                                                        00175
FGO      EQU   *                                                          00180
         EJECT                                                            00185
*                                                                         00205
*              READ IN THE SELECTOR TABLE.                                00210
*                                                                         00215
         OPEN  FILEB                                                      00230
FA       GET   FILEB,RECB                                                 00235
         L     R3,TABTOP        PICK UP ADR. OF PREVIOUS ENTRY.           00245
         LA    R3,7(R3)         ADD FOR THIS ADR.                         00250
         C     R3,TABEND        TABLE EXHAUSTED ?                         00255
         BNL   FB                                                         00260
         ST    R3,TABTOP        IF NOT, STORE CURRENT ENTRY ADR.          00270
         MVC   0(7,R3),RECB     MOVE IN THE ENTRY.
         B     FA                                                         00280
*                                                                         00285
*              TOO MANY SELECTORS.                                        00290
*                                                                         00295
```

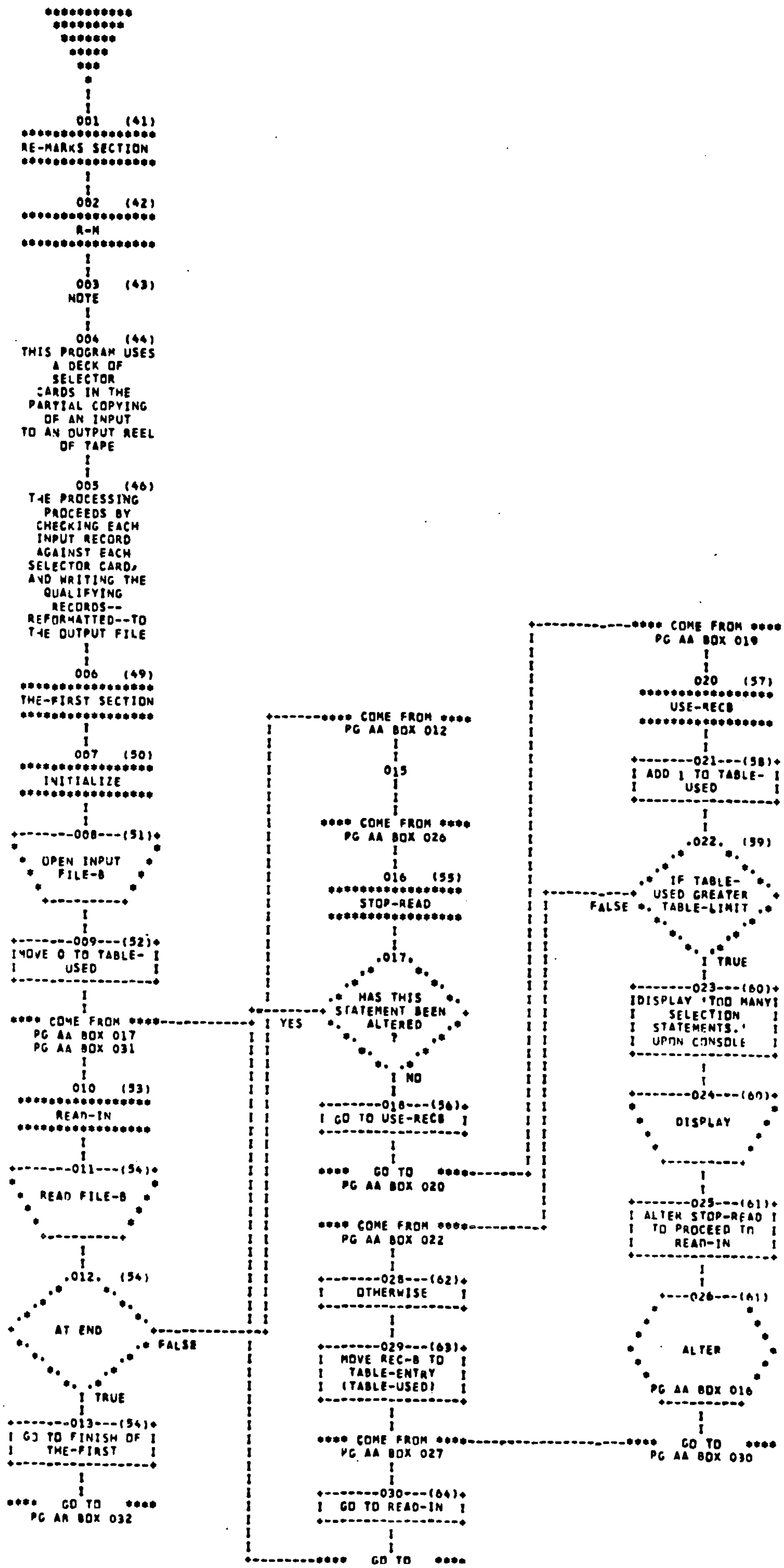
```
         ************
          **********
           ********
            ******
             ***
              *
              I
              I
             001    (1)
          -----------
          / SAMPLE  /
         -----------
              I
              I
          +---002---(1)
         *            *
        *              *
       *  START     0   *
        *              *
         *            *
          +-----------+
              I
              I
             003    (19)
         REGISTER USAGE
              I
              I
             004    (20)
        R.0,1,2,13,14,15-
         -AS DEFINED BY
             IBM.
              I
              I
             005    (21)
         R.10,11--OPEN
          ADDRESSING.
              I
              I
             006    (22)
        R.12--SELECTIVE
          ADDRESSING.
              I
              I
             007    (23)
       R.3,4,5,6,7,8,9--
           SPAN-OF-
          ATTENTION.
              I
              I
             008    (25)
          -----------
         /  FIRST   /
         -----------
              I
              I
     +--------009---(25)+
     ISTM       R14,R12,I
     I     12(R13)      I
     +------------------+
              I
              I
         +---010---(26)
        *             *
       *     BALR      *
      * R10,0 ...SET UP *
       *  OPEN REGS.   *
        *             *
         +-----------+
              I
              I
     +--------011---(28)+
     ILA        R3,4095 I
     I------------------I
     I LA          R11, I
     I   1(R10,R3)      I
     I------------------I
     I MVC         8(4, I
     I R13),=A(*+14)    I
     I------------------I
     IST        R13,*+12I
     +------------------+
              I
              I
         +---012---(33)
        *             *
       *      BAL      *
      * R13,FGO ...SET  *
       *   UP SAVE     *
        *    AREA.    *
         PG AB BOX 015
         +-----------+
              I
              I
             013    (35)
          -----------
         /   FGO    /
         -----------
              I
              I
     ****    GO TO    ****
         PG AB BOX 014
```

B-90


COME FROM PG AA BOX 013
014 (38) READ IN THE SELECTOR TABLE.
COME FROM PG AA BOX 012
015 (40) OPEN FILEB
016 (41) FA
COME FROM PG AB BOX 021
017 (41) GET FILEB, RECB
018 (42) L R3, TABTOP ...PICK UP ADR. OF PREVIOUS ENTRY.
LA R3,7(R3) ...ADD FOR THIS ADR.
C R3, TABEND ...TABLE EXHAUSTED ?
019 (45) BNL FB
EQGRT
LESS
020 (46) ST R3, TABTOP ...IF NOT, STORE CURRENT ENTRY ADR.
MVC 0(7,R3), RECB ...MOVE IN THE ENTRY.
B FA
GO TO PG AB BOX 017
022 (50) TOO MANY SELECTORS.
023 (52) FB
COME FROM PG AB BOX 019
024 (52) PUT MSG ...WRITE MESSAGE TO OPERATOR.
025 (53) FC
026 (53) GET FILEB, RECB ...BYPASS REMAINING SELECTORS.
COME FROM PG AB BOX 029
027 (53) GET
028 (54) B FC
GO TO PG AB BOX 027
COME FROM PG AB BOX 040
041 (70) BE EACH
EQUAL
UNEQ
042 (71) PUT FILEC,RECC ...IF SO, WRITE OUTPUT.
043 (72) B EACH
GO TO PG AB BOX 037
030 (56) CONTROL IS TRANSFERRED HERE FOR EOF ON FILEB.
031 (58) F2
032 (58) CLOSE FILEB
033 (60) COPY SELECTIVELY.
034 (62) SECOND
035 (62) OPEN FILEA, FILEC
036 (63) EACH
COME FROM PG AB BOX 041
PG AB BOX 044
037 (63) GET FILEA, RECA
038 (64) MVC CYEAR, AYEAR ...FORMAT THE OUTPUT RECORD NOW
MVC CMONTH(4),AMONTH ...TO SIMPLIFY DATE-COMPARISON.
MVC CNAME, ANAME
L R15, =A(QUAL) ...USE QUALIFIER SUBROUTINE.
039 (69) BALR R1., R15
040 (69) CP QSW, =P'0' ...SEE IF RECORD WAS SELECTED.
GO TO

S A M P L E   TO DEMONSTRATE COMCHART FOR COBOL   C H A R T / C

```
000000 IDENTIFICATION DIVISION.                                          SAMPLE
000100 PROGRAM-ID. 'SAMPLE'.                                             SAMPLE
000250 ENVIRONMENT DIVISION.                                             SAMPLE
000300 INPUT-OUTPUT SECTION.                                             SAMPLE
000350 FILE-CONTROL.                                                     SAMPLE
000450     SELECT FILE-A   ASSIGN 'FILEA' UTILITY.                       SAMPLE
000500     SELECT FILE-B   ASSIGN 'FILEB' UTILITY.                       SAMPLE
000550     SELECT FILE-C   ASSIGN 'FILEC' UTILITY.                       SAMPLE
000700 DATA DIVISION.                                                    SAMPLE
000750 FILE SECTION.                                                     SAMPLE
010050 FD FILE-A BLOCK 5 RECORDS RECORDING F LABEL RECORDS OMITTED       SAMPLE
010100     DATA RECORD REC-A.                                            SAMPLE
010200 01 REC-A.                                                         SAMPLE
010250     02 CODE-TEST          PICTURE X.                              SAMPLE
010300     02 NAME-FIELD         PICTURE A(20).                          SAMPLE
010350     02 DATE-FIELD.                                                SAMPLE
010400         03 MONTH-FIELD    PICTURE 99.                             SAMPLE
010450         03 DAY-FIELD      PICTURE 99.                             SAMPLE
010500         03 YEAR-FIELD     PICTURE 99.                             SAMPLE
020050 FD FILE-B BLOCK 1 RECORDS RECORDING F LABEL RECORDS OMITTED       SAMPLE
020100     DATA RECORD REC-B.                                            SAMPLE
020200 01 REC-B PICTURE X(7).                                            SAMPLE
030050 FD FILE-C BLOCK 5 RECORDS RECORDING F LABEL RECORDS OMITTED       SAMPLE
030100     DATA RECORD REC-C.                                            SAMPLE
030200 01 REC-C.                                                         SAMPLE
030250     02 DATE-TEST.                                                 SAMPLE
030300         03 YEAR-FIELD     PICTURE 99.                             SAMPLE
030350         03 MONTH-FIELD    PICTURE 99.                             SAMPLE
030400         03 DAY-FIELD      PICTURE 99.                             SAMPLE
030450     02 NAME-FIELD         PICTURE X(20).                          SAMPLE
050050 WORKING-STORAGE SECTION.                                          SAMPLE
050100 77 QUALIFY-SWITCH PICTURE S9 COMPUTATIONAL-3.                     SAMPLE
050150 77 TABLE-LIMIT PICTURE S9(5) COMPUTATIONAL VALUE 0.               SAMPLE
050200 77 TABLE-USED  PICTURE S9(5) COMPUTATIONAL.                       SAMPLE
050250 77  TABLE-X               PICTURE S9(5) COMPUTATIONAL.            SAMPLE
050350 01 THE-TABLE.                                                     SAMPLE
050400     02 TABLE-ENTRY OCCURS 100.                                    SAMPLE
050450         03 TABLE-CODE PICTURE X.                                  SAMPLE
050500         03 TABLE-DATE PICTURE 9(6).                               SAMPLE
100000 PROCEDURE DIVISION.                                               SAMPLE
100100 RE-MARKS SECTION.                                                 SAMPLE
100105 R-M.                                                              SAMPLE
100110     NOTE.                                                         SAMPLE
100115         THIS PROGRAM USES A DECK OF SELECTOR CARDS IN THE PARTIALSAMPLE
100120         COPYING OF AN INPUT TO AN OUTPUT REEL OF TAPE.            SAMPLE
100125         THE PROCESSING PROCEEDS BY CHECKING EACH INPUT RECORD     SAMPLE
100130         AGAINST EACH SELECTOR CARD, AND WRITING THE QUALIFYING    SAMPLE
100135         RECORDS--REFORMATTED--TO THE OUTPUT FILE.                 SAMPLE
101050 THE-FIRST SECTION.                                                SAMPLE
101150 INITIALIZE.                                                       SAMPLE
101200     OPEN INPUT FILE-B.                                            SAMPLE
101250     MOVE 0 TO TABLE-USED.                                         SAMPLE
101350 READ-IN.                                                          SAMPLE
101400     READ FILE-B AT END GO TO FINISH OF THE-FIRST.                 SAMPLE
101450 STOP-READ.                                                        SAMPLE
```


001 (41)
RE-MARKS SECTION
002 (42)
R-M
003 (43)
NOTE
004 (44)
THIS PROGRAM USES A DECK OF SELECTOR CARDS IN THE PARTIAL COPYING OF AN INPUT TO AN OUTPUT REEL OF TAPE
005 (46)
THE PROCESSING PROCEEDS BY CHECKING EACH INPUT RECORD AGAINST EACH SELECTOR CARD, AND WRITING THE QUALIFYING RECORDS--REFORMATTED--TO THE OUTPUT FILE
006 (49)
THE-FIRST SECTION
007 (50)
INITIALIZE
008 (51)
OPEN INPUT FILE-B
009 (52)
MOVE 0 TO TABLE-USED
COME FROM PG AA BOX 017 PG AA BOX 031
010 (53)
READ-IN
011 (54)
READ FILE-B
.012. (54)
AT END
FALSE
TRUE
013 (54)
GO TO FINISH OF THE-FIRST
GO TO PG AA BOX 032
COME FROM PG AA BOX 012
015
COME FROM PG AA BOX 026
016 (55)
STOP-READ
.017.
HAS THIS STATEMENT BEEN ALTERED ?
YES
NO
018 (56)
GO TO USE-RECB
GO TO PG AA BOX 020
COME FROM PG AA BOX 022
028 (62)
OTHERWISE
029 (63)
MOVE REC-B TO TABLE-ENTRY (TABLE-USED)
COME FROM PG AA BOX 027
030 (64)
GO TO READ-IN
GO TO
COME FROM PG AA BOX 019
020 (57)
USE-RECB
021 (58)
ADD 1 TO TABLE-USED
.022. (59)
IF TABLE-USED GREATER TABLE-LIMIT
FALSE
TRUE
023 (60)
DISPLAY 'TOO MANY SELECTION STATEMENTS.' UPON CONSOLE
024 (60)
DISPLAY
025 (61)
ALTER STOP-READ TO PROCEED TO READ-IN
026 (61)
ALTER
PG AA BOX 016
GO TO PG AA BOX 030

E L E M E N T   I N D E X

TYPE	STMT	NAME	REFERENCES IN DECK SEQUENCE
02	14	CODE-TEST	96 E-Q
02	16	DATE-FIELD	
02	26	DATE-TEST	96 E-Q
03	18	DAY-FIELD	(SEE DUPLICATE NAME BELOW)
03	29	DAY-FIELD	
PARA	96	E-Q	
PARA	73	EACH-INPUT	73 EACH-INPUT
SECT	95	EACH-Q	89 Q-B
FD	11	FILE-A	70 INITIALIZE
			73 EACH-INPUT
			83 FINISH
FD	20	FILE-B	50 INITIALIZE
			53 READ-IN
			65 FINISH
FD	23	FILE-C	70 INITIALIZE
			83 FINISH
PARA	65	FINISH	53 READ-IN
PARA	83	FINISH	73 EACH-INPUT
PARA	50	INITIALIZE	
PARA	70	INITIALIZE	
03	17	MONTH-FIELD	(SEE DUPLICATE NAME BELOW)
03	28	MONTH-FIELD	
02	15	NAME-FIELD	(SEE DUPLICATE NAME BELOW)
02	30	NAME-FIELD	
PARA	87	Q-A	
PARA	89	Q-B	89 Q-B
SECT	86	QUALIFIER	73 EACH-INPUT
77	32	QUALIFY-SWITCH	73 EACH-INPUT
			87 Q-A
			89 Q-B
			96 E-Q
PARA	42	R-M	
SECT	41	RE-MARKS	
PARA	53	READ-IN	57 USE-RECB
01	13	REC-A	73 EACH-INPUT
01	22	REC-B	57 USE-RECB
01	25	REC-C	73 EACH-INPUT
PARA	55	STOP-READ	57 USE-RECB
03	38	TABLE-CODE	96 E-Q
03	39	TABLE-DATE	96 E-Q
02	37	TABLE-ENTRY	57 USE-RECB
77	33	TABLE-LIMIT	57 USE-RECB
77	34	TABLE-USED	50 INITIALIZE
			57 USE-RECB
			89 Q-B
77	35	TABLE-X	87 Q-A
			89 Q-B
			96 E-Q
SECT	49	THE-FIRST	
SECT	69	THE-SECOND	

S A M P L E TO DEMONSTRATE COMCHART FOR COBOL C H A R T / C

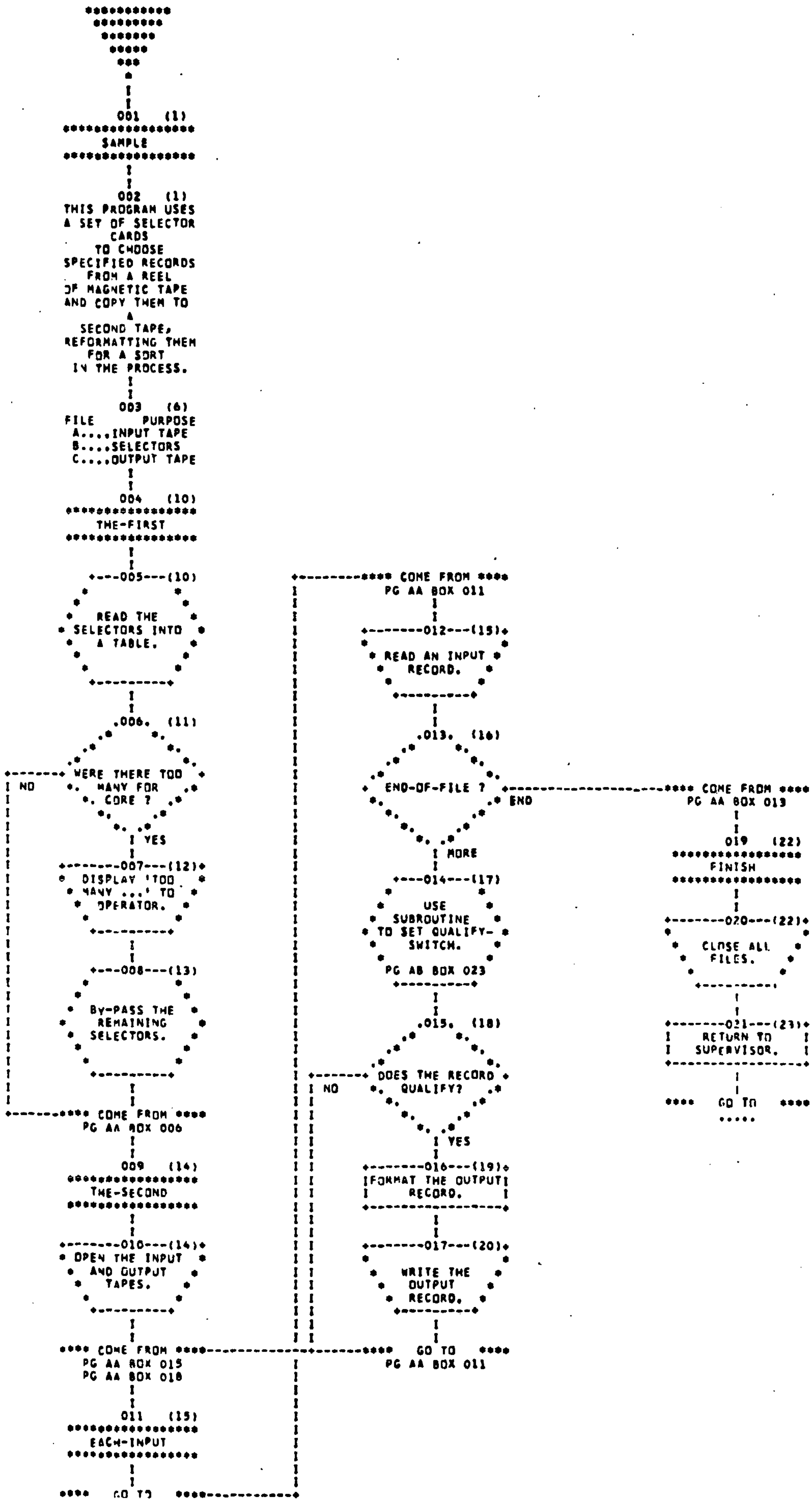
P R O C E D U R E S K E L E T O N

```
PROCEDURES IN DECK SEQUENCE              REFERENCES IN DECK SEQUENCE

 41 RE-MARKS
 42   R-M
 49 THE-FIRST
 50   INITIALIZE
 53   READ-IN                          FROM      57 USE-RECB
                                        TO       65 FINISH
 55   STOP-READ                        FROM      57 USE-RECB
                                        TO       57 USE-RECB
 57   USE-RECB                         FROM      55 STOP-READ
                                        TO       53 READ-IN
                                        TO       55 STOP-READ
 65   FINISH                           FROM      53 READ-IN
 69 THE-SECOND
 70   INITIALIZE
 73   EACH-INPUT                       FROM      73 EACH-INPUT
                                        TO       73 EACH-INPUT
                                        TO       83 FINISH
                                        TO       86 QUALIFIER
 83   FINISH                           FROM      73 EACH-INPUT
 86 QUALIFIER                          FROM      73 EACH-INPUT
 87   Q-A
 89   Q-B                              FROM      89 Q-B
                                        TO       89 Q-B
                                        TO       95 EACH-Q
 95 EACH-Q                             FROM      89 Q-B
 96   E-Q
```

S A M P L E   TO DEMONSTRATE COMCHART FOR LOGIC-DESIGN   C H A R T / L

```
SAMPLE     N                    'THIS PROGRAM USES A SET OF SELECTOR CARDS'V  00005
                                'TO CHOOSE SPECIFIED RECORDS FROM A REEL'  V  00010
                                'OF MAGNETIC TAPE AND COPY THEM TO A'      V  00015
                                'SECOND TAPE, REFORMATTING THEM FOR A SORT'V  00020
                                'IN THE PROCESS.'                             00025
           N               'EDIT.FILE      PURPOSE'                        V  00030
                           'EDIT. A....INPUT TAPE '                        V  00035
                           'EDIT. B....SELECTORS  '                        V  00040
                           'EDIT. C....OUTPUT TAPE'                           00045
THE-FIRST S                     'READ THE SELECTORS INTO A TABLE.'            00105
           D THE-SECOND NO YES 'WERE THERE TOO MANY FOR CORE ?'               00110
           O                    'DISPLAY ''TOO MANY ...'' TO OPERATOR.'       00115
           S                    'BY-PASS THE REMAINING SELECTORS.'            00120
THE-SECOND I                    'OPEN THE INPUT AND OUTPUT TAPES.'            00210
EACH-INPUT I                    'READ AN INPUT RECORD.'                       00220
           D  FINISH END MORE   'END-OF-FILE ?'                               00225
           S QUALIFIER          'USE SUBROUTINE TO SET QUALIFY-SWITCH.'       00230
           D EACH-INPUT NO YES 'DOES THE RECORD QUALIFY?'                     00235
           P                    'FORMAT THE OUTPUT RECORD.'                   00245
           O                    'WRITE THE OUTPUT RECORD.'                    00250
           G EACH-INPUT                                                       00255
FINISH     O                    'CLOSE ALL FILES.'                            00265
           G SUPERVISOR         'RETURN TO SUPERVISOR.'                       00270
           E                                                                  00300
QUALIFIER N                     'THIS SUBROUTINE DETERMINES WHETHER THE'   V  00305
                                'CURRENT RECORD IS SPECIFIED BY THE'       V  00310
                                'SELECTION DECK.'                             00315
           P                    'SET SUBSCRIPT = 0.' 'SET SWITCH OFF'.        00320
REF.Q-A    P                    'SET SUBSCRIPT + 1.'                          00325
           D Q-X YES MORE       'IS TABLE EXHAUSTED ?'                        00330
           S EACH-Q             'USE SUBROUTINE TO MAKE ACTUAL TESTS.'        00335
           D Q-X YES NO         'DID RECORD QUALIFY WITH CURRENT SELECTOR?'   00340
           G REF.Q-A            'CONTINUE.'                                   00345
Q-X        N                    'IF THE RECORD QUALIFIED BY ANY SELECTOR,' V  00355
                                'THE SWITCH HAS BEEN SET ON.'  'OTHERWISE IV  00360
               T IS STILL OFF.'                                               00365
           G EXIT               'EXIT.'                                       00370
EACH-Q     N                    'TO CHANGE SELECTION RULES, JUST CHANGE'   V  00400
                                'THIS ROUTINE.'                               00405
           P                    'IF THE CODE OF THE CURRENT RECORD'        V  00415
                                'MATCHES THE CODE OF THE CURRENT'          V  00420
                                'SELECTOR ENTRY,'                          V  00425
                                'AND THE DATE ON THE RECORD IS ON OR AFTER'V  00430
                                'THE DATE IN THE ENTRY,'                   V  00435
                                'OR IF THE CODE IN THE RECORD IS BLANK,'   V  00440
                                'TURN THE QUALIFY-SWITCH ON.'                 00445
           G EXIT                                                             00465
```

DIAGNOSTIC MESSAGE COUNT   0


001 (1)
SAMPLE
002 (1)
THIS PROGRAM USES A SET OF SELECTOR CARDS TO CHOOSE SPECIFIED RECORDS FROM A REEL OF MAGNETIC TAPE AND COPY THEM TO A SECOND TAPE, REFORMATTING THEM FOR A SORT IN THE PROCESS.
003 (6)
FILE PURPOSE
A....INPUT TAPE
B....SELECTORS
C....OUTPUT TAPE
004 (10)
THE-FIRST
005 (10)
READ THE SELECTORS INTO A TABLE.
.006. (11)
WERE THERE TOO MANY FOR CORE ?
NO
YES
007 (12)
DISPLAY 'TOO MANY ...' TO OPERATOR.
008 (13)
BY-PASS THE REMAINING SELECTORS.
COME FROM PG AA BOX 006
009 (14)
THE-SECOND
010 (14)
OPEN THE INPUT AND OUTPUT TAPES.
COME FROM PG AA BOX 015 PG AA BOX 018
011 (15)
EACH-INPUT
GO TO
COME FROM PG AA BOX 011
012 (15)
READ AN INPUT RECORD.
.013. (16)
END-OF-FILE ?
END
MORE
014 (17)
USE SUBROUTINE TO SET QUALIFY-SWITCH.
PG AB BOX 023
.015. (18)
DOES THE RECORD QUALIFY?
NO
YES
016 (19)
FORMAT THE OUTPUT RECORD.
017 (20)
WRITE THE OUTPUT RECORD.
GO TO PG AA BOX 011
COME FROM PG AA BOX 013
019 (22)
FINISH
020 (22)
CLOSE ALL FILES.
021 (23)
RETURN TO SUPERVISOR.
GO TO

DYNACHART

Application Programming Company

GENERAL

DYNACHART accepts COBOL programs as input and produces a flowchart and diagnostic messages as well as optional individual listings of the source program, and cross references within the source program's Data Division only on both the Data and Procedure Divisions.

DYNACHART can be implemented on any computer system capable of supporting a COBOL compiler and configured with at least 30K characters of core, one disc or four tape units, one card reader, and one line printer. DYNACHART is written in a minimum subset of COBOL.

It leases for $4,400 the first year. Maintenance use charge for subsequent years is $700 annually.

Package Output

Source Program Listing - Each line consists of an 80-80 image of the source card preceded by a generated line number.

Flowchart - Detailed flowchart in same logical order as the program source code. Many features under user's control.

Label Cross Reference - Labels are alphabetically listed as are defining source code sequence number and all statements referencing label.

Note: DYNACHART is one of the most expensive documentation packages. Its output features are not so unique as to justify the cost of the package.

EZFLOW

Systonetics Corporation

GENERAL

EZFLOW accepts FORTRAN source programs, reformats and renumbers statements, derives a cross-reference list between statement labels and references of original and new program versions, produces a statement number reference table that shows use and location of numbers on restored flowcharts, and produces a new source deck listing. The package uses a default option control card when none of the option control cards are needed.

The package runs on IBM 360/30 and up (OS), with 110K bytes of core storage; CDC 6000 Series (SCOPE), with 32K bytes of core storage. Peripherals include a card reader and line printer, and optionally a disc or tape. The source language of this package is FORTRAN IV.

EZFLOW is available in two versions off-the-shelf: for the CDC 6000 series and for the IBM 360 or 370 series. The price is $3,500 for a three-year use license as a one-time charge.

PACKAGE OUTPUT

Flowchart - This is a logic flowchart of the restored deck. This is a single column chart of rectangular blocks and diamond-shaped boxes, one source code statement is equal to one block on the chart.

Input and Output Source Deck - The package punches and prints a copy of the output source deck and prints a listing of the input deck.

Cross-Reference List - This listing of statement labels and references shows the old and resequenced source code label numbers and the source list line number.

Statement Reference Table - This table summarizes the program flow by giving the use and location of numbers that are in parentheses in the flowchart.

Note: EZFLOW has automatic conversion to and from the BCD and EBCDIC character sets.

```
* - * - *  E Z F L O W  * - * - *       PROCESSING ROUTINE - -  MAIN

 1        SUBROUTINE SAMPLE (  A   , B ,CC ,C,* , * ,    D        )
 2  C==================================================================
 3  C           THIS IS AN EXAMPLE TO DEMONSTRATE SOME OF THE FEATURES
 4  C         OF E Z F L O W
 5  C==================================================================
 6  C          NOTE THAT STATEMENTS OF THE SAME TYPE ARE ALL GROUPED TOGETHER
 7              DIMENSION A (10 ),  Z(  5),  X X Y Z  (1000 ) ,WXYZ(  3 )
 8          DIMENSION              Q(50)  ,LABE L (10),  ENDO  O( 2 0 )
 9        DOUBLE PRECISION   B(20)
10        COMPLEX *  16 C ( 20 )
11           REAL *  8Q
12            DATALABEL/ '  THIS IS A SAMPLE OF A DATA STATEMENT    '   /
13                       EQUIVALENCE    ( A ,B )   ,      (C ,  D )
14  C           THIS IS AN EXAMPLE OF A MULTIPLE RETURN ROUTINE
15   95                CALL       SUB2  ( A,B  ,  &2   ,   &20,INDEX)
16          CALL   SUB1   (   A,B,  6HLABEL    ,' LABEL 2  ',  &2)
17  C           EXAMPLE OF A COMPUTED GO TO  STATEMENT
18               GOTO    (51,63,2,4,2,84,95,84),  INDEX
19  C                 EXAMPLE OF DO LOOP PROCESSING
20  51           D O      152         I    =  M    ,    NOFDO
21          A  (I)  =     Z(   I)
22        WXYZ(  I)=    I
23    152    ENDDO  ( I)=( I+  1)*A(  I)
24  C             EXAMPLE OF ARITHMETIC IF STATEMENT
25  84        I F ( ENDDO ( 10)-123456*SIN( FLOAT( INDEX)))1000,2000,4000
26  C          EXAMPLE OF READ STATEMENT
27  20           READ( 5, 5000,END=2,ERR=1000)        A
28  5000       FORMAT   (     10E8.2)
29  4      W RITE (6   ,   8041    )   A
30  8041  FOR MAT  (   '1    INPUT DATA  A   '// ( 10E8.2))
31  2                RETURN         1
32  C/      THIS COMMENT CARD WILL START A NEW PAGE  (SLASH IN COLUMN 2)
33  C       EXAMPLE OF A DO LOOP TERMINATING IN A LOGICAL IF STATEMENT
34   63         DO 7 7   K     =     1   ,  50
35                      A         ( K  )     =K     *    K*A   (  K  )
36  77     IF    (MOD (K,10) .EQ . 0  )  READ(5,5000, END =2,ERR= 4000) C,D
37                        GOT     O        8
38  C       EXAMPLE OF ANOTHER TYPE OF MULTIPLE RETURN STATEMENT
39  4000     CALL     SUB3(A,  B),RETURNS(2,  20)
40            GOTO50
41  2000    C O N T I N U E
42  8        D O 7J    = M ,  N
43  7        A   (J)=        J
44  1000   R ETURN                    2
45  50     WRITE    (    6,78    )    A
46  78        FORMAT      (  'A='E18.5)
47                                    RETURN
48        EN                          D
```

```
* - * - *  E Z F L O W  * - * - *       PROCESSING ROUTINE - -  SAMPLE
```

N E W - O L D S T A T E M E N T N U M B E R C R O S S R E F E R E N C E

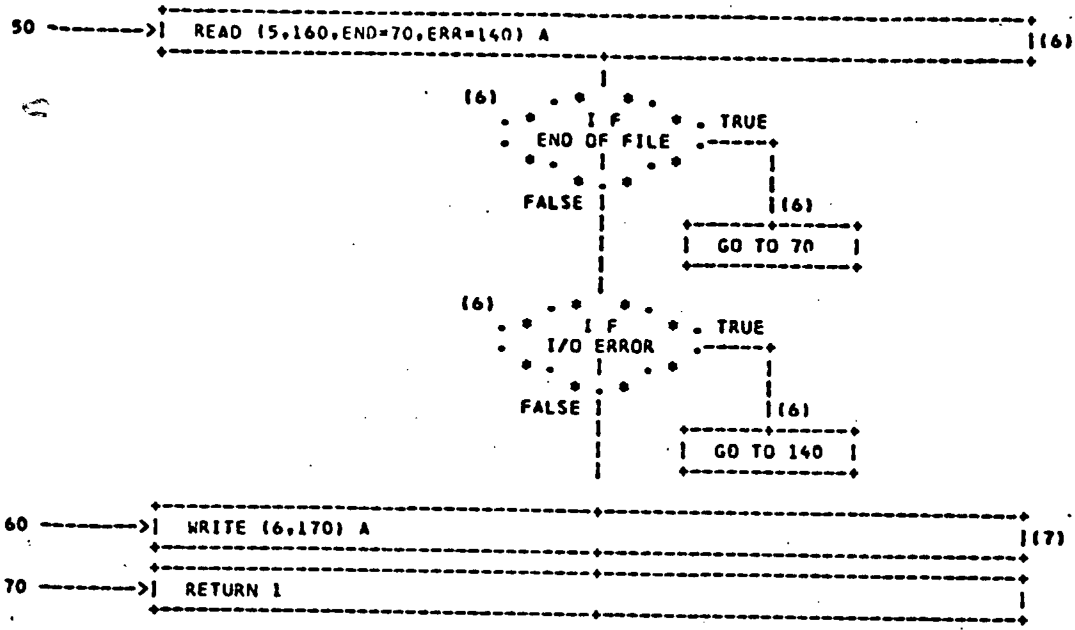
NEW	OLD	LINE	OLD	NEW	LINE
10	95	15	2	70	31
20	51	20	4	60	29
30	152	23	7	130	43
40	84	25	8	120	42
50	20	27	20	50	27
60	4	29	50	150	45
70	2	31	51	20	20
80	63	34	63	80	34
90	77	36	77	90	36
100	4000	39	78	180	46
110	2000	41	84	40	25
120	8	42	95	10	15
130	7	43	152	30	23
140	1000	44	1000	140	44
150	50	45	2000	110	41
160	5000	28	4000	100	39
170	8041	30	5000	160	28
180	78	46	8041	170	30

* - * - * E Z F L O W * - * - * PROCESSING ROUTINE - - SAMPLE

```
      SUBROUTINE SAMPLE (A,B,CC,C,*,*,D)
C======================================================================
C     THIS IS AN EXAMPLE TO DEMONSTRATE SOME OF THE FEATURES
C     OF E Z F L O W
C======================================================================
C     NOTE THAT STATEMENTS OF THE SAME TYPE ARE ALL GROUPED TOGETHER
      DIMENSION A(10),Z(5),XXYZ(1000),WXYZ(3)
      DIMENSION Q(50),LABEL(10),ENDDO(20)
      DOUBLE PRECISION B(20)
      COMPLEX*16 C(20)
      REAL*8 Q
      DATA LABEL/'  THIS IS A SAMPLE OF A DATA STATEMENT    '/
      EQUIVALENCE (A,B),(C,D)
C     THIS IS AN EXAMPLE OF A MULTIPLE RETURN ROUTINE
10    CALL SUB2 (A,B,&70,&50,INDEX)
      CALL SUB1 (A,B,6HLABEL ,' LABEL 2  ',&70)
C     EXAMPLE OF A COMPUTED GO TO  STATEMENT
      GO TO (20,80,70,60,70,40,10,40), INDEX
C     EXAMPLE OF DO LOOP PROCESSING
20    DO 30 I=M,NOFDO
      A(I)=Z(I)
      WXYZ(I)=I
30    ENDDO(I)=(I+1)*A(I)
C     EXAMPLE OF ARITHMETIC IF STATEMENT
40    IF (ENDDO(10)-123456*SIN(FLOAT(INDEX))) 140,110,100
C     EXAMPLE OF READ STATEMENT
50    READ (5,160,END=70,ERR=140) A
60    WRITE (6,170) A
70    RETURN 1
C/    THIS COMMENT CARD WILL START A NEW PAGE  (SLASH IN COLUMN 2)
C     EXAMPLE OF A DO LOOP TERMINATING IN A LOGICAL IF STATEMENT
80    DO 90 K=1,50
      A(K)=K*K*A(K)
90    IF (MOD(K,10).EQ.0) READ (5,160,END=70,ERR=100) C,D
      GO TO 120
C     EXAMPLE OF ANOTHER TYPE OF MULTIPLE RETURN STATEMENT
100   CALL SUB3 (A,B),RETURNS(70,50)
      GO TO 150
110   CONTINUE
120   DO 130 J=M,N
130   A(J)=J
140   RETURN 2
150   WRITE (6,180) A
      RETURN
C
160   FORMAT (10E8.2)
170   FORMAT ('1    INPUT DATA   A    '//(10E8.2))
180   FORMAT ('A='E18.5)
      END
```

```
                                +-----------------------------------------------------------------+
                                | SUBROUTINE SAMPLE (A,B,CC,C,*,*,D)                               |
                                +-----------------------------------------------------------------+
                                                                  I
                                 =================================================================
                                                                  I
                                                                  I
                                     THIS IS AN EXAMPLE TO DEMONSTRATE SOME OF THE FEATURES
                                                                  I
                                                                  I
                                     OF E Z F L O W
                                                                  I
                                                                  I
                                 =================================================================
                                                                  I
                                                                  I
                                     NOTE THAT STATEMENTS OF THE SAME TYPE ARE ALL GROUPED TOGETHER
                                                                  I
                                +-----------------------------------------------------------------+
                                | DIMENSION A(10),Z(5),XXYZ(1000),WXYZ(3)                          |
                                | DIMENSION Q(50),LABEL(10),ENDDO(20)                              |
                                +-----------------------------------------------------------------+
                                +-----------------------------------------------------------------+
                                | DOUBLE PRECISION B(20)                                           |
                                +-----------------------------------------------------------------+
                                +-----------------------------------------------------------------+
                                | COMPLEX*16 C(20)                                                 |
                                +-----------------------------------------------------------------+
                                +-----------------------------------------------------------------+
                                | REAL*8 Q                                                         |
                                +-----------------------------------------------------------------+
                                +-----------------------------------------------------------------+
                                | DATA LABEL/'  THIS IS A SAMPLE OF A DATA STATEMENT `  '/         |
                                +-----------------------------------------------------------------+
                                +-----------------------------------------------------------------+
                                | EQUIVALENCE (A,B),(C,D)                                          |
                                +-----------------------------------------------------------------+
                                                                  I
                                     THIS IS AN EXAMPLE OF A MULTIPLE RETURN ROUTINE
                                                                  I
                                +-----------------------------------------------------------------+
             10 -------->| CALL SUB2 (A,B,&70,&50,INDEX)                                    |(1)
                                +-----------------------------------------------------------------+
                                             +---------------------------------------+
                                             | I F     SUBROUTINE RETURNS            |(1)
                                             | RETURN =1      ,2                     |
                                             | GO TO   70     ,50                    |
                                             +---------------------------------------+
                                +-----------------------------------------------------------------+
                                | CALL SUB1 (A,B,6HLABEL ,' LABEL 2  ',&70)                        |(2)
                                +-----------------------------------------------------------------+
                                             +---------------------------------------+
                                             | I F     SUBROUTINE RETURNS            |(2)
                                             | RETURN =1                             |
                                             | GO TO   70                            |
                                             +---------------------------------------+
                                                                  I
                                    EXAMPLE OF A COMPUTED GO TO  STATEMENT
                                                                  I
                                     +------------------------------------------------------+
                                     | I F              COMPUTED GO TO ON  INDEX              |(3)
                                     | INDEX = 1    ,2    ,3    ,4    ,5    ,6    ,7    ,8    |
                                     | GO TO   20   ,80   ,70   ,60   ,70   ,40   ,10   ,40   |
                                     +------------------------------------------------------+


                                    EXAMPLE OF DO LOOP PROCESSING

                   ........                                   .............
         20----->.  DO  .(4)                                  .           .
      +----<----. 30    ..................................... I=M,NOFDO .
      I             ........                                  ......+......
      I                         +-----------------------------------------------------------------+
      I                         | A(I)=Z(I)                                                        |
      I                         | WXYZ(I)=I                                                        |
      +--------> 30 -------->| ENDDO(I)=(I+1)*A(I)                                              |
                                +-----------------------------------------------------------------+
                                                                  I
                                    EXAMPLE OF ARITHMETIC IF STATEMENT
                                                                  I
             40 -------->---->---->---->---->---->---->---->-I
                                            (5)           .  *   *  .
                                         +------------ *      I F      * -----------+
                                         | (ENDDO(10)-123456+SIN(FLOAT(INDEX)))   |
                                         |------------ * .              . * -----------|
                                         I                        * . *                I
                                         I                          I                  I
                                     MINUS I                     ZERO I              PLUS I
                                     +--+--+                     +--+--+             +--+--+
                                     |GO TO|                     |GO TO|             |GO TO|
                                     | 140 |                     | 110 |             | 100 |
                                     +-----+                     +-----+             +-----+
```

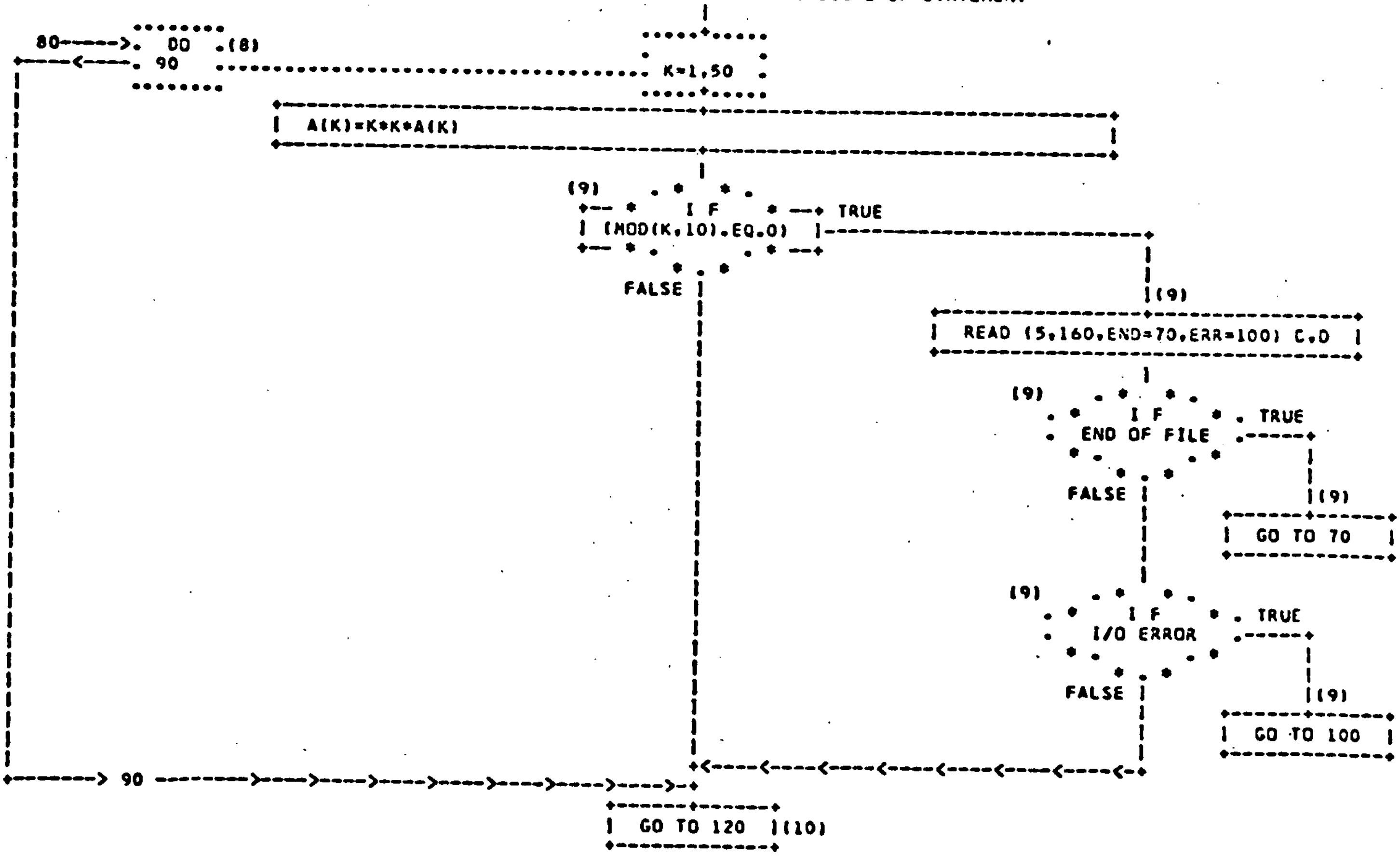
EXAMPLE OF READ STATEMENT



* - * - * E Z F L O W * - * - * PROCESSING ROUTINE - - SAMPLE

/ THIS COMMENT CARD WILL START A NEW PAGE (SLASH IN COLUMN 2)

EXAMPLE OF A DO LOOP TERMINATING IN A LOGICAL IF STATEMENT



EXAMPLE OF ANOTHER TYPF OF MULTIPLE RETURN STATEMENT

```
                   +--------------------------------------------------------+
                   |                                                        |(11)
100 -------->|  CALL SUB3 (A,B),RETURNS(70,50)                              |
                   +--------------------------------------------------------+
                              +-----------------+---------------+
                              | I F     SUBROUTINE RETURNS      |(11)
                              | RETURN =1      ,2               |
                              | GO TO    70    ,50              |
                              +---------------------------------+
                                         +-------+-------+
                                         |  GO TO 150    |(12)
                                         +---------------+

                   +--------------------------------------------------------+
                   |                                                        |
110 -------->|  CONTINUE                                                    |
                   +--------------------------+-----------------------------+
                  ........                ....+....
120---->.  DO   .(13)                     .       .
+----<----. 130  ......................... J=M,N .
|         ........                        ....+....
|                  +--------------------------+-----------------------------+
|                  |                                                        |
+------> 130 -------->|  A(J)=J                                             |
                   +--------------------------+-----------------------------+
                   +--------------------------+-----------------------------+
                   |                                                        |
140 -------->|  RETURN 2                                                    |
                   +--------------------------+-----------------------------+
                   +--------------------------+-----------------------------+
                   |                                                        |(14)
150 -------->|  WRITE (6,180) A                                             |
                   +--------------------------+-----------------------------+
                   +--------------------------+-----------------------------+
                   |                                                        |
                   |  RETURN                                                |
                   +--------------------------+-----------------------------+
                   +--------------------------+-----------------------------+
                   |                                                        |
160 -------->|  FORMAT (10E8.2)                                             |
170 -------->|  FORMAT ('1   INPUT DATA  A   '//(10E8.2))                   |
180 -------->|  FORMAT ('A='E18.5)                                          |
                   +--------------------------+-----------------------------+
                   +--------------------------+-----------------------------+
                   |                        E N D                           |
                   |                                                        |
                   +--------------------------------------------------------+
```

* - * - * E Z F L O W * - * - * PROCESSING ROUTINE - - SAMPLE

STATEMENT	* - * - * B L O C K (S) NUMBER TYPE	R E F E R E N C I N G * - * - * NUMBER TYPE	NUMBER TYPE	NUMBER TYPE	NUMBER TYPE
10 -	3 GO				
20 -	3 GO				
30 -	4 DO				
40 -	3 GO ,	3 GO			
50 -	1 CALL ,	11 CALL			
60 -	3 GO				
70 -	1 CALL , 9 IF I/O ,	2 CALL , 11 CALL	3 GO ,	3 GO ,	6 I/O
80 -	3 GO				
90 -	8 DO				
100 -	5 IF ,	9 IF I/O			
110 -	5 IF				
120 -	10 GO				
130 -	13 DO				
140 -	5 IF ,	6 I/O			
150 -	12 GO				
160 -	6 I/O ,	9 IF I/O			
170 -	7 I/O				
180 -	14 I/O				

FACTS

Bonner and Moore Associates, Incorporated

GENERAL

FACTS accepts FORTRAN source programs as input, analyzes programs and subroutines, and produces cross-referenced information according to the option selection by the programmer.

The package runs on the Sigma 7 computer.

PACKAGE OUTPUT

Source Listing - Source statements of the program are automatically listed.

Program Reports - Six program reports are generated:

(1) Common Report

(2) Local Report

(3) Format Statement

(4) Statement Label Report

(5) Recap

(6) Global Report

```
                    **WRTAPE  (04)** /LCOM    / COMMON REPORT                                  B-106
                                                                                                PAGE
********************************************************************************************************
        VARIABLE          *                                  APPEARANCES
                          *
********************************************************************************************************
                 *        *                    *                           *
      NAME       * TYPE   *   DECLARATIVE      *      DEFINITIONAL         *          REFERENCE
                 *        *                    *                           *
********************************************************************************************************
   NV1           *   I    *  0005              * 0017 0021                 * 0037
   NV2           *   I    *  0005              * 0019                      * 0017 0019 0023 0037
```

NAME	TYPE	DECLARATIVE	DEFINITIONAL	REFERENCE
All variables named in blank common are printed alphabetically. Each variables type is indicated by one of the following one-character codes.	<u>C</u>omplex <u>D</u>ouble precision <u>I</u>nteger <u>L</u>ogical <u>R</u>eal	FACTS generated statement numbers in which the blank common variable name appears in a declarative manner; (dimension, equivalence statements, etc.) by four digit sequence numbers.	FACTS generated statement numbers in which variable name is defined (as in left hand side of arithmetic statement).	Four digit FACTS generated st ment numbers in which the bla common name is referenced (as right hand side of arithmetic statement).

Figure 6-1. Common Report

```
                              **WRTAPE  (04)** LOCAL REPORT                                PAGE 00(
*********************************************************************************************************
      VARIABLE           *                              APPEARANCES
                         *
*********************************************************************************************************
                    *         *                    *                            *
     NAME           * TYPE    *   DECLARATIVE      *        DEFINITIONAL        *          REFERENCE
                    *         *                    *                            *
*********************************************************************************************************
 I                  *  I      *                    * 0027 0029 0032 0034 0040   * 0027 0029 0032 0034 0040
 IUNIT              *  I      *                    *                            * 0001 0015 0048
 JFILE              *  I      *                    * 0039                       * 0041
```

NAME	TYPE	DECLARATIVE	DEFINITIONAL	REFERENCE
All variables named in blank common are printed alphabetically. Each variables type is indicated by one of the following one-character codes.	Com- plex Dou- ble pre- ci- sion Inte- ger Logi- cal Real	FACTS generated statement numbers in which the blank common variable name appears in a declarative manner; (dimension, equivalence statements, etc.) by four digit sequence numbers.	FACTS generated statement numbers in which variable name is defined (as in left hand side of arithmetic statement).	FACTS generated statement numbers in which the blank common name is referenced (as in right hand side of arithmetic statement).

Figure 6-2. Local Report

WRTAPE (04) FORMAT STATEMENT PAGE 1

FORMAT LABEL	APPEARANCES: DEFINITION	APPEARANCES: REFERENCE
901	0011	0037
902	0012	0040
903	0013	0043
Sequential list of format numbers used in the subroutine or program.	FACTS generated statement number in which format statement label is defined.	FACTS generated statement number of the instructions which reference the format statement label listed in the first column.

Figure 6-3. Format Statement Report

```
                    **WRTAPE  (04)** STATEMENT LABEL REPORT                    PAGE 00
******************************************************************************************
              *
              *                              APPEARANCES
   STATEMENT  *
     LABEL    ****************************************************************************
              *                                  *
              *          DEFINITION              *              REFERENCE
              *                                  *
******************************************************************************************
      10      *  0014                            *  0050
     100      *  0026                            *  0025
              *                                  *
```

Statement label	Definition	Reference
Sequential list of all statement numbers used in the subroutine or program.	FACTS generated statement number in which the format statement label is defined.	FACTS generated statement number of the instructions which reference the statement listed in the first column. column.

Figure 6-4. Statement Label Report

B-110

```
                                         **XXXX    (03)*RECAP                                          PAGE 00
********************** ********************************************************************** ************************
                     * *                                                                    * *
  ROUTINE ENTRYS     * *                    INTERNAL FUNCTIONS                              * *   EXTERNAL NAMES
                     * *                                                                    * *
********************** ********************************************************************** ************************
              *      * *              *                         *                           * *                *
   NAME       *  AT  * *    NAME      *         DEFINED         *        REFERENCED         * *     NAME       *    AT
              *      * *              *                         *                           * *                *
********************** ********************************************************************** ************************
XXXX    (03)  * 0011 * * F       (03) * 0004                    * 0008                      * * A       (01)   * 0005
YYYY    (03)  * 0006 * *              *                         *                           * * IABS    (01)   * 0010
              *      * *              *                         *                           * * Y       (01)   * 0007

Name and location    * ************Self explanatory********************* *All references in this
of all subroutines   * *                                                  *program or function to
or function entry    * *                                                  *external subroutines o
points, or if a pro-* *                                                   *functions.
gram, the name       * *                                                  *
specified in the     * *                                                  *
PC2 name field,      * *                                                  *
with the location    * *                                                  *
blend.               * *                                                  *
```

Figure 6-5. Recap Report

```
                                                        GLOBAL REPORT                                            PAGE 0
*************************************************************************************************************************
                                     *                                    *
          PROGRAM ENTRYS             *                                    *
                                     *                                    *
*************************************         THIS PROGRAM CALLS          *          THIS PROGRAM CALLED BY
                   *                 *                                    *
    PRIMARY        *   SECONDARY     *                                    *
                   *                 *                                    *
*************************************************************************************************************************
 MAIN              *                 *   IABS     (01)    RDTAPE   (04)   *
                   *                 *   WRTAPE   (04)    XX       (02)   *
                   *                 *   XXXX     (03)                    *
 WRTAPE    (04)    * RDTAPE    (04)  *   IABS     (01)    IOERR    (01)   *    MAIN
 XX        (02)    *                 *                                    *    MAIN
 NNN               *                 *                                    *
                   *                 *                                    *
                   *                 *                                    *

 List of every*program entry *All external references for the   *All programs or subroutines of
 point defined*in the entire *programs or subroutines listed in *run which call subroutines or
 run.         *              *PRIMARY.                          *functions listed under
 Entry points *Entry point   *                                  *PROGRAM ENTRYS.
 defined via  *defined via   *                                  *
 the PC2 name *the entry     *                                  *
 field, sub-  *statements.   *                                  *
 routine or   *              *                                  *
 function     *              *                                  *
 statement    *              *                                  *
```

Figure 6-6. Global Report

FLOWGEN/F

CalComp

GENERAL

FLOWGEN/F accepts FORTRAN source cards and produces ink-on-page flowcharts. It generates plot commands to drive a CalComp Plotter. It gives no other listings about the program than the flowchart.

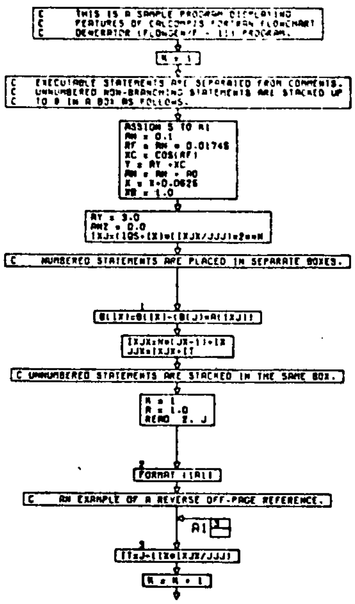
PACKAGE OUTPUT

Flowchart - This is a detailed flowchart that is given on a ink-on-paper CalComp Plotter. Comments are enclosed in boxes.

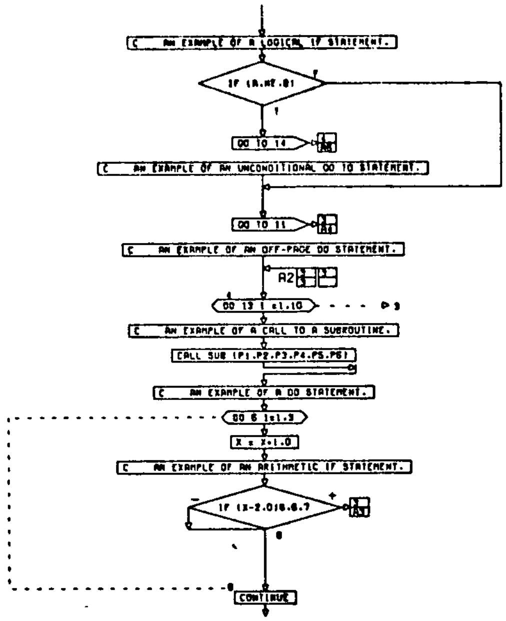
FLOWGEN/F SAMPLE PROGRAM

```
C *    THIS IS A SAMPLE PROGRAM TO DISPLAY
C *    FEATURES OF THE CALCOMP - FORTRAN - FLOWCHART GENERATOR
C * * *
       DIMENSION DATA (1024)
C EXECUTABLE STATEMENTS ARE SEPARATED FROM COMMENTS
C * UNNUMBERED NON-BRANCHING STATEMENTS ARE STACKED UP TO
C *      8 IN A BOX AS FOLLOWS
       AN = 0.1
       RF = AN * 0.01745
       XC = COS (RF)
       Y = AY +XC
       AN = AN + AD
       X = X+0.0625
       XB = 1.0
       AY = 3.0
         ANZ = 0.0
C LOGICAL -IF- STATEMENTS ARE AS FOLLOWS
 1     IF (A.NE.B) RETURN
       GO TO 5
C THE PREVIOUS STATEMENT REPRESENTS A -GO TO- STATEMENT
C CALLS TO SUBROUTINES ARE AS FOLLOWS - CONTINUATION CARD SHOWN ALSO
       CALL SUB (P1,P2,P3
     1 P4,P5,P6)
C A -DO LOOP- IS SHOWN IN THE FOLLOWING MANNER
 5     DO 6 I=1,3
       X = X+1.0
       IF (X-2.0)6,6,7
 6     CONTINUE
C AN ARITHMETIC -IF- WAS SHOWN IN THE ABOVE -DO LOOP-
C COMPUTED AND ASSIGNED -GO TO - ARE REFLECTED AS FOLLOWS
 7     GO TO (10,5,1),K
C NUMBERED STATEMENTS ARE PLACED IN SEPARATE BOXES
 10    K = 1
C UNNUMBERED STATEMENTS ARE STACKED IN THE SAME BOX
       K = 1
       A = 1.0
       READ 12, J
 12    FORMAT (1A1)
C THIS IS AN OFF-PAGE DO-LOOP EXAMPLE
       DO 13 I =0, 1
       X = X+1.0
 13    CONTINUE
C RETURN AND STOP AND END ARE AS FOLLOWS
       RETURN
       STOP
C AFTER THE FOLLOWING END STATEMENT IS DRAWN AN -UNFULFILLED
C        REFERENCE- STATEMENT IS LISTED ON THE PRINTER. ANY
C        STATEMENT NUMBERS OR THE WORD NONE WILL THEN BE LISTED.
       END
```

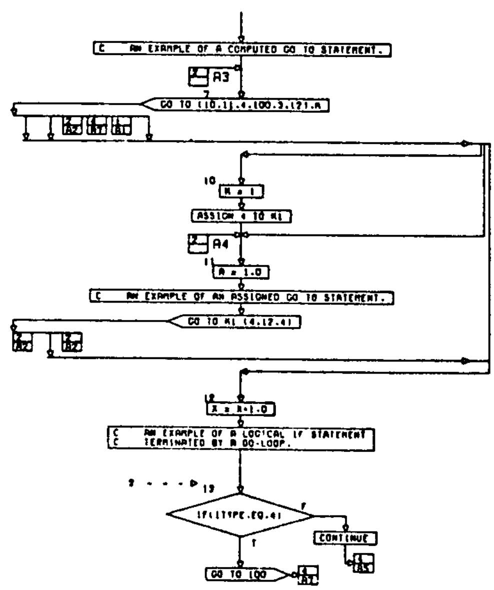
C.3



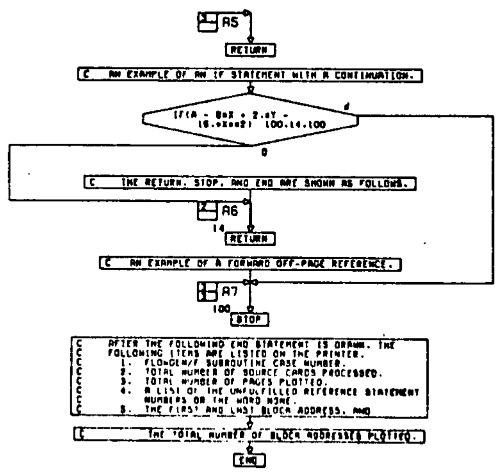
CONT. ON PG 2



CONT. ON PG 3



CONT. ON PG 4



FORFLOW

DNA System, Inc.

GENERAL

FORFLOW accepts FORTRAN source decks as input. It is a two-program system which consists of two distinct programs: FLOWA produces a flowchart of the source program and SEQ generates a source listing containing re-sequenced FORTRAN statement numbers in ascending order by fives, and changes all branch and formatted input/output statement numbers to agree with the new statement numbers.

The package can operate on an IBM 1130 computer under control of the monitor system or on an IBM 1800 under TSX version 3 or MTX. Minimum machine requirements for the 1130 include 8K bytes of core, 1 card reader, 1 line printer, and 1 disc. For 1800 a minimum of 10K bytes of variable core is required. The package itself is written in FORTRAN and includes some assembler subroutines.

The package costs $480.00 for the card system and $600.00 for disc system.

PACKAGE OUTPUT

Flowchart - at some level of detail as the source program. All flowchart blocks are rectangular.

Resequenced listing - this can be a printed listing of the resequenced source deck, or it can be of punched cards.

FORTRAN VARIABLE NAME DOCUMENTER

Data for Management Decision

GENERAL

The FORTRAN Variable Name Documenter accepts FORTRAN source code as input. The package generates a sequenced listing of a FORTRAN program, a numerical list of the statement numbers used in that program, and an alphabetical list of variable names in the program.

The package is designed for any user's configuration that is capable of supporting a FORTRAN compiler. Auxiliary storage is provided by one disc. The package consists of two programs and a sort. Source language is FORTRAN.

Purchase price is $350 for the object deck or $500 for the source deck.

PACKAGE OUTPUT

Source Listing - a listing of the FORTRAN source code.

Statement Number Listing - a numeric listing of the program statement numbers.

Variable Name Listing - an alphabetic listing of program variable names.

QUICK-DRAW

National Computer Analysts, Incorporated

GENERAL

QUICK-DRAW translates source programs written in Assembler, AUTOCODER, COBOL, PL/I, and FORTRAN into flowcharts and related cross-references. It also provides a diagnostic check list and modified instruction list to aid with debugging.

The system runs on IBM 360; Burroughs 25/35/55; HIS 200, 400, 600, 800; ICL 1900; RCA Spectra 70 or 3301; and Univac 9400/1100. System supervision may be OS, TOS, DOS, MCP, DAPS, MODZ, TDOS, or EXEC 8. Minimum core storage is 32K bytes for DOS and TOS and 65K bytes for OS.

Leases are $1,900 to $6,300 depending on version (3-year lease) and $1,400 to $2,100 a year depending on version.

PACKAGE OUTPUT

Listings for Assembly:

Flowchart - This is a detailed flowchart of the program. It contains all the ASSEMBLY statements in logical groups placed in boxes whose shape is determined by the type of statement.

Cross-Reference by Term - This listing contains entries for all terms used within the program.

Cross-Reference to Equator - This listing repeats the definition and references to equated tags in same format as in the cross-reference by term.

Source Program Listing - Shows the contents of each card read.

Supplementary Listing - This list consists of the following:

(1) Diagnostic Check List

(2) Unrecognized Op-Codes

(3) Assembler Directing Op-Codes

(4) Input-Output Type of-Codes

(5) Perform Type Instruction

(6) Path Terminations

(7) Privileged Instructions

(8) Program Linkage Op-Code-Entered here are ENTRY, EXTRN, etc.

(9) User Macros and Macro Calls

(10) Program Segmentation Op-Codes

Listings for FORTRAN:

Source Program Listing - Shows contents of each card.

Flowchart - This is a two-dimentional flowchart. Flowcharts may be in a double-page format.

Statement Label Cross-Reference - This listing contains all statement numbers belonging to procedural statements or referenced by procedural statements.

<u>Data Name Cross Reference</u> - This contains all data field names, subroutine names, and labels used in the program and lists, by card number, references made to those names.

<u>Diagnostic Check List</u> - This provides a number of diagnostic printouts for those statements with error.

<u>Listing for COBOL</u>:

<u>Source Statement Listing</u> - Shown five digit card number used for card identification, plus the card contents.

<u>Flowchart</u> - A detailed flowchart of the program is given.

<u>Cross-Reference by Term</u> - This listing contains entries for data-names, procedure-names, literals, figurative constants. Each entry contains the term and all references to that term.

<u>Diagnostic Flow Summary</u> - This is a graphic summary of the structure of the program in flowchart form. It contains all paragraph and section names together with the range of card numbers comprising each paragraph, their locations on the flowchart and notes and diagnostic pertaining to each paragraph.

Note: All versions of the flowcharts are at the same level of detail as the source code. The COBOL flowchart only, gives the option of compressing the level of detail to achieve a higher level flowchart.

QUICK-DRAW is designed to select a convenient branch point whenever possible to end a page.

QUICK-DRAW is second only to AUTOFLOW in its number of installations.

```
01087                                   14.01
MVC   NAMFLG(1),ONEP
LA    5,UPDATE
CALCULATE AVERAGE BALANCE

01089  14.02
02.15
B      PRINTRTN

TABLE3DP
        E-22.36
01090  14.03
CP
*BWADP)UN(1),ONEP BE*     BE
GETEND25

01092                                   14.04
CP    9(2,7),NINE50
LA    5,TABLE2

01094  14.05
* BL      PRINTRTN *     BL
                         02.15
                         PRINTRTN

01095                                   14.06
LA    5,GLNAMAD

01096  14.07
* BE      PRINTRTN *     BE
                         02.15
                         PRINTRTN

01097                                   14.08
LA    5,NEWACC1

010 8
```

```
01108  14.15
CP
MASWRK+57(7),
*ZEROP(7) IS BALANCE* BE
ZERO OE TESTAFF IF YES*
THERE HAVE BEEN NO*
ENTRIES TO NOW*
A/C

01110                                   14.16
MVC   MASWRK+122(2),DDD
TODAYS DATE=DATE OF START OF
PERIOD

01111  14.17
CP
MASWRK+56(1),
*NINEP DO NOT UPDATE* BE
AVERAGE BALANCE BF
TESTAFF FIELD FOR
GENERAL LEDGER*

01113                                   14.18
MVC   MASWRK+124(5),MASWRK+58
MOVE CURRENT BALANCE INTO
MVN   MASWRK+128(1),MASWRK+63
AVERAGE BALANCE FIELD

TESTAFF
E-04.26 E-14.14     E-14.15 E-14.17
01115  14.19
TM
*MASWRK+40,X'02'*
IS IT A STAFF      * BZ
*ACCOUNT BZ TABLE4 IF NO*
BRANCH
                         14.31
                         TABLE4

UDATEL
01117                                   14.20
MVC   LOBALNCE,MASWRK+58
BAL. OF STAFF A/C
MVN   LOBALNCE+4(1),MASWRK+63
MOVE SIGN
```

```
COMPLO2
                    E-14.25
01129                                   14.28
CP    LOBALNCE(5),MASWRK+124(5)
COMPARE CURRENT AND AV BAL
BNL TABLE4 IF CURRENT BALANCE
HIGH OR EQUAL

01129  14.29
* SEE ABOVE *      BNL

01131                                   14.30
MVC   MASWRK+124(5),LOBALNCE
OTHERWISE UPDATE FIELD

TABLE4
E-14.19 E-14.24    E-14.27 E-14.29
                   E-27.18 E-27.24
01132                                   14.31
LA    5,BACK

01133                                   14.32
/              COMRG              /

01134                                   14.33
MVC   MLTAREA2(2),3(1)
MM
MVC   MLTAREA2+2(2),0(1)
DD
PACK  CALC(3),MLTAREA2(4)
MVO   MONTH(3),CALC(3)

01138  14.34
CLI
*MONTH,X'09' BNH*     BNH
STRQUEST

01140                                   14.35
XR    6,6
IC    6,MONTH
SH    6,SIXH
STC   6,MONTH

STRQUEST
```

B-121

```
01099                 I
                      V               14.10
   * TXNS FOR SBWA DP SUSPENSE
ACCOUNT HAVE ALREADY BEEN
PROCESSED. IF ANY RECORD ON
CARC25 AFTER THIS MUST BE
ERROR RECORDS TO BE PUT TO THE
DP SUSPENSE ACCOUNT. OTHERWISE
WILL GO TO EOF

GETEND25
                      I E-14.03 E-14.12 V
                      I<----------------
01102                 V             14.11 A
*************************************
H          BAL   6,ADHARFWD         H
H      ( 29.34) ADHARFWD            H
*************************************

          01103 V  14.12
                   CP
       *  MASWRK(4),5(4,7) * BE
    *COMPARE ACCOUNT NUMBERS*---->*
       *    BE GETEND25   *

          01105 V  14.13
               *******
               * 22.34 *
               *******
                B      TESTERR

TABLE3A
E-13.25 E-28.08 I E-28.10
          01106 V  14.14
                   CP
            * MASWRK+122(2),*
        * ZEROP+5(2) IS THIS* BNE
   * THE FIRST DAY TXNS FOR*-------*
       *    BNE TESTAFF   *

                  I
                  V
          (NEXT COLUMN)
```

```
         * CAD6A+5,C'F' *
       * WAS STAFF INTEREST* BE
     * CALCULATED TODAY BE  *---->*
       *COMPIO IF NO BRANCH*

          01121 V  14.22
                  CP
BL
*<----* MASWRK+57(7),ZEROP(7) *
          *  BL PUTZERO   *

01123             V                 14.23
*-----------------------------------*
I   MVC   MASWRK+124(5),LOBALNCE    I
I    OTHERWISE UPDATE FIELD ON      I
I              CARI2                I
*-----------------------------------*

          01124 V  14.24
               *******
               * 14.31 *
               *******
                B      TABLE4

COMPLO
                  I E-14.21
                  I<---------------
          01125 V  14.25
                   CP
        * MASWRK+57(7),ZEROP* BNL
      * IS ACCOUNT OVERDRAWN *--------*
        * BNL COMPLO2 IF NO *
               BRANCH

PUTZERO
                  I E-14.22
*--------------->I
01127             V                 14.26
*-----------------------------------*
I   MVC   MASWRK+124(5),ZEROP+2     I
I   OTHERWISE PUT ZEROS IN FIELD    I
*-----------------------------------*

          01128 V  14.27
               *******
               * 14.31 *
               *******
                     TABLE4
```

```
       *MASWRK+40,X'80'*
     *STATEMENT REQUESTED* BZ
   *BZ STATED BRANCH IF OFF*--*
                              15.21
                              STATED

          01146 V  14.37
               *******
               * 16.19 *
               *******
                B      STMNTRTN

DAYRTN
E-15.08 E-15.40 I E-15.41 E-17.38
                I E-17.39
01147           V                   14.38
*-----------------------------------*
I     MVC   DAOFWEEK,ZEROX          I
I     OI     CAD6A+5,X'40'          I
*-----------------------------------*

          01149 V  14.39
                  CLI
           *CAD6A+5,C'A' NO*   BCR
        * STATEMENTS PRINTED*---*
           * TODAY BCR 8,5 *
                                5

          01151 V  14.40
                  CLI             BE
           *CAD6A+5,C'U' BE*------*
             *  ONETOIT  *
                                15.09
                                ONETOIT

          01153 V  14.41
                  CLI             BE
           *CAD6A+5,C'V' BE*------*
             *  TWOTOIT  *
                                15.11
                                TWOTOIT
                I
                V
          (NEXT PAGE)
```

B-122

SAMPLE CROSS REFERENCE BY TERM

```
02/06/72             CROSS REFERENCE BY TERM OF     SEF FIRST TITLE,START OR CSECT FOR STANDARD                    PAGE  13
*****************************************************************************************************************************
* LITERAL OR TAG AND INCREMENT    * LINE *    REFERENCING LINE OPERAND NUMBER AND OP-CODE                                   *
*****************************************************************************************************************************
*                                 *      *
* OLDCA        DS   10CL115       * 2489 * 0378 1 MVC    0380 1 MVC    0733 1 MVC    0763 1 MVC    0813 1 MVC    0891 1 MVC    *
*                                 *      * 1335 1 MVC    1518 1 MVC    1588 1 MVC    2033 1 MVC    2062 1 MVC    2180 1 MVC    *
*                                 *      * 2376 1 MVC    2374 1 MVI    0373 2 MVC    0378 2 MVC    1500 2 MVC    2178 2 MVC    *
*                                 *      * 2369 2 MVC    2375 2 MVC    0016 3 DTFSD                                            *
*                   +1            *      * 0379 1 MVC    2375 1 MVC                                                            *
*                   +1024         *      * 1522 1 MVC    1592 1 MVC    2190 1 MVC    0377 2 MVC    1504 2 MVC    2182 2 MVC    *
*                                 *      * 2373 2 MVC                                                                          *
*                   +13           *      * 0603 1 MVC    0589 1 MVI    0590 2 MVC                                              *
*                   +14           *      * 0590 1 MVC                                                                          *
*                   +173          *      * 0385 1 MVC    0765 1 MVC    0817 1 MVC    1333 1 MVC    2380 1 MVC    1313 2 MVC    *
*                   +177          *      * 0766 1 MVC    0818 1 MVC    1334 1 MVC                                              *
*                   +184          *      * 0767 1 MVC    0819 1 MVC    1318 2 MVC                                              *
*                   +256          *      * 1519 1 MVC    1589 1 MVC    2187 1 MVC    0374 2 MVC    1501 2 MVC    2179 2 MVC    *
*                                 *      * 2370 2 MVC                                                                          *
*                   +4            *      * 0381 1 MVC    0730 1 MVC    0764 1 MVC    0816 1 MVC    0988 1 MVC    1336 1 MVC    *
*                                 *      * 2030 1 MVC    2059 1 MVC    2377 1 MVC    0715 2 LA     0974 2 LA     2020 2 LA     *
*                                 *      * 2047 2 LA     0717 2 MVC    0976 2 MVC    0985 2 MVC    1315 2 MVC    2005 2 MVC    *
*                                 *      * 2048 2 MVC                                                                          *
*                   +439          *      * 0768 1 MVC    0820 1 MVC                                                            *
*                   +440          *      * 1319 2 MVC                                                                          *
*                   +512          *      * 1520 1 MVC    1590 1 MVC    2188 1 MVC    0375 2 MVC    1502 2 MVC    2180 2 MVC    *
*                                 *      * 2371 2 MVC                                                                          *
*                   +694          *      * 0769 1 MVC    0821 1 MVC                                                            *
*                   +696          *      * 1320 2 MVC                                                                          *
*                   +768          *      * 1521 1 MVC    1591 1 MVC    2189 1 MVC    0376 2 MVC    1503 2 MVC    2181 2 MVC    *
*                                 *      * 2372 2 MVC                                                                          *
*                   +8            *      * 0382 1 MVC    2378 1 MVC    1997 2 CP     2257 2 CP     1283 2 MVC    0631 2 UNPK   *
*                                 *      * 0708 2 UNPK   0965 2 UNPK   2108 2 UNPK   2213 2 UNPK                               *
*                   +9            *      * 2171 1 CP     0222 2 CP     0282 2 CP     0319 2 CP     0616 2 CP     1255 2 CP     *
*                                 *      * 1527 2 CP     2193 2 CP                                                             *
*                   +949          *      * 0770 1 MVC    0822 1 MVC                                                            *
*                   +952          *      * 1321 2 MVC                                                                          *
*                                 *      *
* OLDTONEW                        * 2178 * 0342 1 B      0620 1 B      1693 1 B      2097 1 B      2247 1 B      2261 1 B      *
*                                 *      * 2263 1 B      0224 1 BH     0287 1 BH     0321 1 BH     1332 1 BH     2174 1 BL     *
*                                 *      * 2246 2 LA                                                                           *
*                                 *      *
* ONE          DC   F'1'          * 2581 * 0864 2 A      1395 2 A      1401 2 A      1404 2 A      1407 2 A      1410 2 A      *
*                                 *      * 1413 2 A      1453 2 A      1458 2 S      1466 2 S                                  *
*                                 *      *
* ONEH         DC   H'1'          * 2478 * 0592 2 AH                                                                           *
*                                 *      *
* ONEP         DC   P'1'          * 2526 * 0136 2 AP     0151 2 AP     0395 2 AP     0578 2 AP     0593 2 AP     0622 2 AP     *
*                                 *      * 0714 2 AP     0805 2 AP     0812 2 AP     0847 2 AP     0866 2 AP     0868 2 AP     *
*                                 *      * 0902 2 AP     0909 2 AP     0928 2 AP     0935 2 AP     1040 2 AP     1048 2 AP     *
*                                 *      * 1347 2 AP     1428 2 AP     1433 2 AP     2034 2 AP     2063 2 AP     2210 2 AP     *
*                                 *      * 2295 2 AP     2327 2 AP     0162 2 CP     0462 2 CP     1073 2 CP     1090 2 CP     *
*                                 *      * 1257 2 CP     1761 2 CP     1348 2 CP     1492 2 CP     1494 2 CP     1496 2 CP     *
*                                 *      * 1545 2 CP     1609 2 CP     1665 2 CP     1689 2 CP     1694 2 CP     1820 2 CP     *
*                                 *      * 1825 2 CP     1833 2 CP     1837 2 CP     1908 2 CP     2143 2 CP     2167 2 CP     *
*                                 *      * 2149 2 CP     2244 2 CP     2248 2 CP     2253 2 CP     2255 2 CP     2340 2 CP     *
*                                 *      * 2345 2 CP     2347 2 CP     0313 2 MVC    0368 2 MVC    0737 2 MVC    0776 2 MVC    *
*                                 *      * 0872 2 MVC    1087 2 MVC    1671 2 MVC    1708 2 MVC    1721 2 MVC    1742 2 MVC    *
*                                 *      * 1872 2 MVC    1835 2 MVC    1839 2 MVC    2091 2 MVC    2101 2 MVC    2344 2 MVC    *
*                                 *      * 1435 2 MVZ    1436 2 MVZ    0101 2 SP     0106 2 SP                                 *
*                                 *      *
* ONETOIT                         * 1168 * 1152 1 BE                                                                           *
*                                 *      *
* ONLYCA                          * 2244 * 2170 1 BE     2256 1 BE                                                             *
*                                 *      *
* ONLYCR                          * 2038 * 2014 1 BE     2036 2 LA                                                             *
*                                 *      *
* ONLYENT                         * 2340 * 2254 1 BE     2348 1 BE                                                             *
*                                 *      *
* ONLYMAS                         * 2248 * 2168 1 BE     2346 1 BE                                                             *
*                                 *      *
* OOOH                            * 0199 * 0196 1 BL                                                                           *
*                                 *      *
* OPRALPH      DC   C                 * 2559 * 0141 2 MVC                                                                          *
*                   *8            *      * 0145 2 MVC                                                                          *
*                                 *      *
* ORAC                            * 0644 * 0702 1 BH     0704 1 BNL                                                            *
*                                 *      *
* ORTN                            * 1476 * 1194 1 BE                                                                           *
*                                 *      *
* OTHERAC                         * 1600 * 1551 2 BC                                                                           *
*                                 *      *
* OUTHERE                         * 2385 * 2082 1 BE                                                                           *
*                                 *      *
* OUTPUT                          * ---- * 0024 4 DTFSD 0042 4 DTFSD                                                           *
*                                 *      *
* PATTERN      EQU  PATTN         * 2660 *                                                                                     *
*                   +PATTN        *      * 0263 2 MVC    0386 2 MVC    0633 2 MVC    0943 2 MVC    1019 2 MVC    1613 2 MVC    *
*                                 *      * 1653 2 MVC    1655 2 MVC    1746 2 MVC    1865 2 MVC    1866 2 MVC    1890 2 MVC    *
*                                 *      * 1891 2 MVC    1975 2 MVC    1980 2 MVC    2067 2 MVC    2311 2 MVC                  *
*                                 *      *
* PATTN        DC   X             * 2462 * 2660 1 EQU                                                                          *
*                                 *      *
* PLANE                           * 1938 * 1925 1 BL                                                                           *
*                                 *      *
* PLANEPR      DC   C'_MAYFAIR_'  * 2551 * 1938 2 MVC                                                                          *
*                                 *      *
* PREVTXP                         * 1582 * 1578 1 BNE                                                                          *
*                                 *      *
* PRINTAC                         * 1871 * UNREFERENCED
```

B-124

CROSS REFERENCE TO EQUATES

This listing repeats the definition and references to equated tags in the same format as in the CROSS REFERENCE BY TERM.

```
6/72          CROSS REFERENCE TO EQUATES OF       SEE FIRST TITLE,START OR CSECT FOR STANDARD

*******************************************************************************************************
TERAL OR TAG AND INCREMENT   * LINE *   REFERENCING LINE OPERAND NUMBER AND OP-CODE
*******************************************************************************************************
                             *      *
CKSTAF     EQU BKSTAF        * 2655 *
               =BKSTAF       *      * 0560 1 BNL
                             *      *
AMVT T     EQU WAMVTO        * 2659 *
               =WAMVTD       *      * 1771 1 BE
                             *      *
LSHT       EQU FSFILE+8      * 2650 *
               =FSFILE       *      * 0805 1 AP     0812 1 AP     0758 1 MVC    0761 1 MVC    0800
                             *      * 1304 1 MVC    1305 1 MVC    1306 1 MVC    1309 1 MVC    1311
4          EQU 14            * 0043 * 0116 1 BR     0114 1 LM     0064 1 STM
5          EQU 15            * 0044 * 0065 2 LR
           EQU 2             * 0031 * 0102 1 BAL    0109 1 BAL    0112 1 BAL    0144 1 BR
           EQU 3             * 0032 * UNREFERENCED
           EQU 4             * 0033 * 0108 1 BASE   0111 1 LA     0094 1 LM     0107 2 BASE   01
           EQU 5             * 0034 * 0096 1 BASE   0146 1 BASE   0094 2 LM
           EQU 6             * 0035 * UNREFERENCED
           EQU 7             * 0036 * UNREFERENCED
           EQU 8             * 0037 * 0065 1 LR     0063 2 USING
           EQU 9             * 0038 * UNREFERENCED
*******************************************************************************************************
```

SPECIAL CROSS REFERENCE

This listing repeats entries from the CROSS REFERENCE BY TERM if they are one of the following:

- Non-Branch References to Instruction Tags. This will call attention to modified instructions, such as switches and also serve as a check for the "UNENTERED STATEMENT" diagnostic, where in fact, the statement was entered by means of a load address and a branch register.
- Branches to Data.
- Anything unusual that we feel should be highlighted, such as ENTRY and EXTRN references.

```
6/72          SPECIAL CROSS REFERENCES FOR       QUICK-DRAW SAMPLE PROGRAM A-12 FOR ASSEMBLY

*******************************************************************************************************
TERAL OR TAG AND INCREMENT   * LINE *   REFERENCING LINE OPERAND NUMBER AND OP-CODE
*******************************************************************************************************
                             *      *
RSTIME                       * 0072 *
            +1               *      * 0073 1 MVI
                             *      *
LOW                          * 0146 * 0138 3 PRTOV
                             *      *
INTER                        * 0153 * 0136 1 CNTRL 0138 1 PRTOV 0142 1 PUT
            +22              *      * 0074 1 MVI
4           EQU 14           * 0043 * 0116 1 BR
                             *      *
            EQU 2            * 0031 * 0144 1 BR
                             *      *
LOWSW                        * 0098 *
            +1               *      * 0087 1 MVI    0090 1 MVI
*******************************************************************************************************
```

PROGRAM SEGMENTATION OP-CODES

CSECT and START are in this section.

SAMPLE SUPPLEMENTARY LISTING

```
SUPPLEMENTARY LISTINGS FOR        QUICK-DRAW SAMPLE PROGRAM A-12 FOR ASSEMBLY

    DIAGNOSTIC CHECK LIST          *00146  UNENTERED STATEMENT
                                   *00162  UNENTERED STATEMENT
    PROGRAM LINKAGE OP-CODES       *00061  ENTRY BIB095
                                                 -FOR ACCESS BY COBOL.
    ASSEMBLER DIRECTING OP-CODES   *00003  SPACE
                                                 10
                                   *00023  EJECT
                                   *00045  SPACE
                                                 2
                                   *00051  EJECT
                                   *00052  SPACE
                                                 12
                                   *00060  SPACE
                                                 2
                                   *00063  USING *,R8
                                                 ESTABLISH ADDRESSABILITY
                                   *00148  EJECT
                                   *00149  SPACE
                                                 2
                                   *00152  SPACE
                                                 2
                                   *00161  EJECT
                                   *00167  EJECT
                                   *00168  SPACE
                                                 3
                                   *00169  LTORG
    INPUT_OUTPUT TYPE OP-CODES     *00136  CNTRL PRINTER,SP,1
                                                 ADVANCE ONE LINE IMMEDIATELY.
                                   *00138  PRTOV PRINTER,12,OFLOW
                                                 TEST FOR OVERFLOW
                                   *00142  PUT   PRINTER,(R4)
                                                 ADDRESS OF COBOL-LINE IN GPR
                                                 4.
                                   *00153  DTFPR DEVADDR=SYSLST,
                                                 PRINTER ON SYSTEM LOGICAL
                                                 UNIT.
                                                 IOAREA1=ALINE1,IOAREA2=ALINE2,
                                                 WORKA=YES,
                                                 SPECIFIED IN GPR 4.
                                                 BLKSIZE=132,PRINTOV=YES,
                                                 TEST FOR CHANNEL-12.
                                                 CONTROL=YES
                                                 CNTRL MACRO USED FOR
                                                 FORMS-CONTROL.
                                   *00162  PRMOD IOAREA2=YES,WORKA=YES,
                                                 PRINTOV=YES,CONTROL=YES
    PERFORM TYPE INSTRUCTIONS      *00102  BAL   R2,PRINT
                                                 ...PRINT AS SET UP BY COBOL-
                                                 ADVANCE 1.
                                   *00109  BAL   R2,PRINT
                                                 AND PRINT CAPPED COBOL LINE.
                                   *00112  BAL   R2,PRINT
                                                 PRINT UNDERLINE.
```

Reference Format Listing (Source Program Listing)

The entire source program is listed as QUICK-DRAW reads it for processing.

The complete contents of each card is printed, including the card sequence number found in columns 73-80. In addition, a four-digit statement sequence number, assigned by QUICK-DRAW, is printed to the left of the card image. This sequence number will appear on the flowchart above the top left corner of the corresponding symbol.

The heading of each page is labeled "Reference Format Listing." The QUICK-DRAW Release Number is printed following the last source statement.

```
*0044*           IBIG=0
*0045*           MOST=0
*0046*           K=0
*0047*           INDEX8=1
*0048*           J=(JHIGH-1)/8+1
*0049*           DO 54 I=1,J
*0050*           DO 53 L=1,8
*0051*           NTII (L)=0
*0052*       53  NTJI (L)=0
*0053*       54  WRITE(8'INDEX8,20) (NTII(M),NTJI(M),M=1,8)
*0054*           WRITE(3,501)JHIGH
*0055*      501  FORMAT(1H0,I4,19H NODES SET TO ZERO.)
*0056*           INDEX7=1
*0057*        6  READ(7'INDEX7,5)NODI,NODJ,ITIME,COST
*0058*        5  FORMAT(3I5,F6.0)
*0059*           IF(NODI-9999)7,11,7
*0060*        7  INDEX8=(NODI-1)/8+1
*0061*           READ(8'INDEX8,20) (NTII(M),NTJI(M),M=1,8)
*0062*           INDEX8=INDEX8-1
*0063*           M=8-(INDEX8*8-NODI)
*0064*           NTII(M)=1
*0065*           WRITE(8'INDEX8,20) (NTII(M),NTJI(M),M=1,8)
*0066*           INDEX8=(NODJ-1)/8+1
*0067*           READ(8'INDEX8,20) (NTIJ(N),NTJJ(N),N=1,8)
*0068*           INDEX8=INDEX8-1
*0069*           N=8-(INDEX8*8-NODJ)
*0070*           NTJJ(N)=1
*0071*           WRITE(8'INDEX8,20) (NTIJ(N),NTJJ(N),N=1,8)
*0072*           TCOST=TCOST+COST
*0073*           MOST=MOST+ITIME
*0074*           K=K+1
*0075*           IF(NODI-IBIG)9,9,8
*0076*        8  IBIG=NODI
*0077*        9  IF(NODJ-IBIG)6,6,10
*0078*       10  IBIG=NODJ
*0079*           GO TO 6
*0080*       11  KLAST=K
*0081*           WRITE(3,502)
*0082*      502  FORMAT(1H0,36HFLAGS ALL SET FOR NODE NUMBER CHECK.)
*0083*           IRST=0
*0084*           LAST=0
*0085*           K=IBIG
*0086*       55  INDEX8=(K-1)/8+1
*0087*           READ(8'INDEX8,20) (NTII(M),NTJI(M),M=1,8)
*0088*           M=8-((INDEX8-1)*8-K)
*0089*           IF(NTII(M))12,13,12
*0090*       12  IF(NTJI(M))14,15,14
```

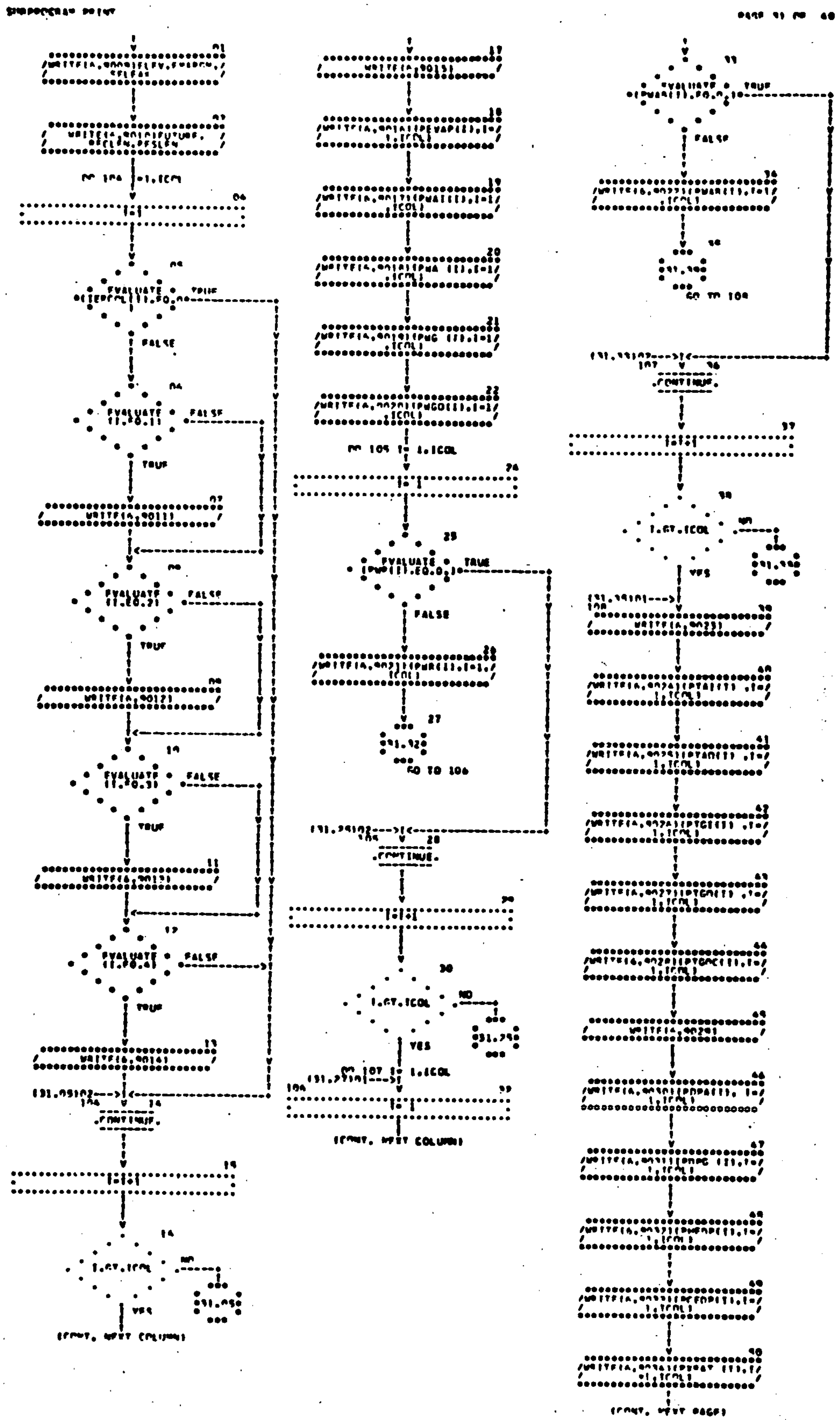


Figure 1 - Sample Portion of Flowchart

Statement Label Cross Reference Listing

The first column of this listing contains all statement numbers belonging to procedural statements or referenced by procedural statements. These statement numbers are listed in numeric order. In addition, Column 1 contains the names of any subroutines referenced by the program.

The second column lists the page and symbol box numbers where each statement number or subroutine appears on the flowchart. These location references are in the form "pp.bb", where pp is the flowchart page number and bb is the symbol box number on that page. If a statement number is undefined, Column 2 contains the entry "UNDEF". If a referenced subroutine is not included in the source program, Column 2 contains the entry "EXTRN". Format statement labels will have the entry "FORMAT" in this column.
The third column lists all references to the statement number or subroutine name. These references are also in the form "pp.bb". If a statement number is not referenced, it is so indicated.

Each subprogram included in the source deck has its own cross-reference listing, beginning on a separate page. If any statement number appears more than once in the main program or in the same subprogram, it is flagged as MULTIPLY-DEFINED.

STATEMENT LABEL CROSS REFERENCE

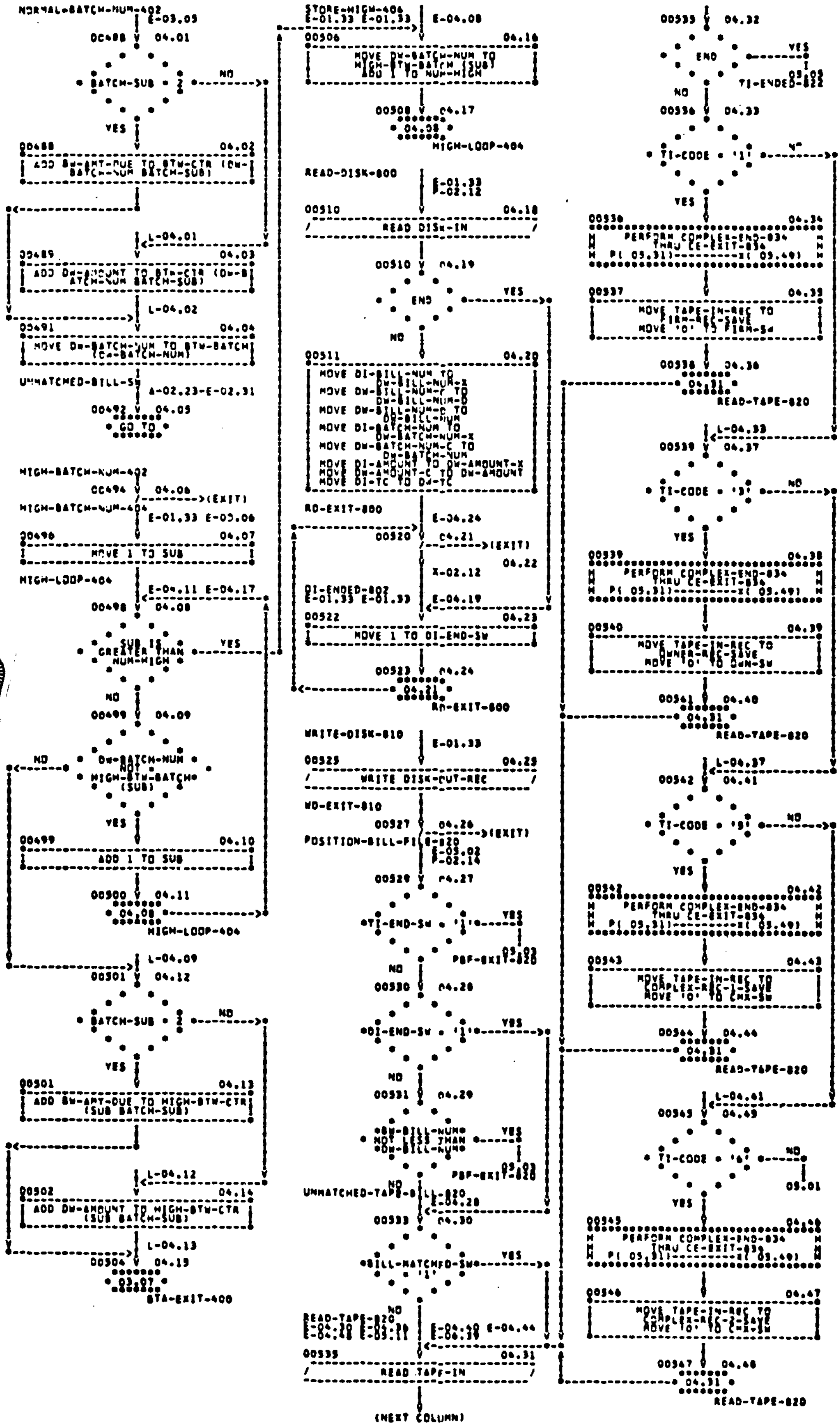
52	01.01	04.02	06.24		
53	01.17	01.15			
54	01.20	01.14			
55	01.44	02.09			
60	04.03	03.32	04.32		
63	04.33	04.30			
64	05.02	05.01			
65	05.27	05.26			
66	06.05	04.45	05.34	05.36	06.03
67	06.16	NOT REFERENCED			
71	03.18	03.17			
80	06.12	05.01			

3.4 Data Name Cross Reference Listing

This table contains all data field names, subroutine names, and labels used in the program and lists the card numbers in which references are made to those names or labels.

This listing is not available from the FORTRAN compiler and offers a distinct advantage to programmers for debugging, maintenance and documentation purposes. Very often a quick glance through this listing is sufficient to detect keypunching errors or incorrect references.

```
                                 NCA  F O R T R A N  QUICK DRAW SAMPLE

*******************************************************************************
  *                                                CROSS REFERENCE BY CARD NUMBER
*******************************************************************************
  * 0076
  *
  * 0017
  *
  * 0010
  *
  * 0137    0150    0159    0168    0174    0200
  *
  * 0108    0117    0119    0121    0130    0132    0198    0205    0213
  * 0231    0233    0241    0242
  *
  * 0076    0077
  *
  * 0133    0169    0201    0205    0226    0231
  *
  * 0135    0170    0203    0205    0220    0228    0231
  *
  * 0133    0172    0204    0205    0222    0223    0225    0231
  *
  * 0134    0170    0202    0205    0220    0221    0227    0231
  *
  * 0077
  *
  * 0058    0085    0127    0132
  *
  * 0056    0087    0121    0129    0132    0152
  *
  * 0057    0083    0121    0122    0128    0132    0151
  *
  * 0092
  *
  * 0003    0019    0046    0049    0052    0084    0085    0101    0161
  * 0194    0201
  *
  * 0059    0090    0123    0124    0126    0132    0153
  *
```


NORMAL-BATCH-NUM-402
BATCH-SUB = 2
ADD BW-AMT-DUE TO BTW-CTR (DW-BATCH-NUM BATCH-SUB)
ADD DW-AMOUNT TO BTW-CTR (DW-BATCH-NUM BATCH-SUB)
MOVE DW-BATCH-NUM TO BTW-BATCH (CW-BATCH-NUM)
UNMATCHED-BILL-SW
GO TO
HIGH-BATCH-NUM-402
HIGH-BATCH-NUM-404
MOVE 1 TO SUB
HIGH-LOOP-404
SUB IS GREATER THAN NUM-HIGH
DW-BATCH-NUM NOT = HIGH-BTW-BATCH (SUB)
ADD 1 TO SUB
BATCH-SUB = 2
ADD BW-AMT-DUE TO HIGH-BTW-CTR (SUB BATCH-SUB)
ADD DW-AMOUNT TO HIGH-BTW-CTR (SUB BATCH-SUB)
BTA-EXIT-400
STORE-HIGH-406
MOVE DW-BATCH-NUM TO HIGH-BTW-BATCH (SUB) ADD 1 TO NUM-HIGH
HIGH-LOOP-404
READ-DISK-800
READ DISK-IN
END
RD-EXIT-800
DI-ENDED-802
MOVE 1 TO DI-END-SW
WRITE-DISK-810
WRITE DISK-OUT-REC
WD-EXIT-810
POSITION-BILL-FI
TI-END-SW = '1'
PBF-EXIT-820
DI-END-SW = '1'
BILL-MATCHED-SW = '1'
UNMATCHED-TAPE-BILL-820
READ-TAPE-820
READ TAPE-IN
(NEXT COLUMN)
TI-ENDED-822
TI-CODE = '1'
PERFORM COMPLEX-END-834 THRU CE-EXIT-834
MOVE TAPE-IN-REC TO FIRM-REC-SAVE MOVE '0' TO FIRM-SW
READ-TAPE-820
TI-CODE = '3'
MOVE TAPE-IN-REC TO OWNER-REC-SAVE MOVE '0' TO OWN-SW
TI-CODE = '5'
MOVE TAPE-IN-REC TO COMPLEX-REC-1-SAVE MOVE '0' TO CMX-SW
TI-CODE = '6'
READ-TAPE-820

SAMPLE CROSS REFERENCE BY TERM

'9'	LITERAL	T 0542-004.61
'0'	LITERAL	T 0543-004.63
HIGH-VALUES	FIG-CON	S 0385-001.43 S 0565-005.12
NUMERIC	FIG-CON	T 0367-001.31
RUN	FIG-CON	H 0668-006.22
SPACE	FIG-CON	S 0353-001.24
SPACES	FIG-CON	S 0386-001.45 S 0418-002.17 S 0470-003.01 S 0585-005.35 S 0634-006.08 S 0675-006.26 S 0688-006.30
ZERO	FIG-CON	T 0421-002.24 T 0468-002.42 S 0656-006.17
ZEROS	FIG-CON	S 0369-001.32 S 0377-001.38 S 0612-005.48 S 0613-005.48 S 0614-005.48 S 0615-005.48 S 0616-005.48 S 0617-005.48 S 0618-005.48 S 0619-005.48 S 0620-005.48 S 0621-005.48 T 0687-006.35
0	LITERAL	S 0631-006.06 T 0642-006.12 T 0672-006.25
01	LITERAL	T 0336-001.03 T 0340-001.08 T 0344-001.13 T 0349-001.18
02	LITERAL	T 0337-001.05 T 0341-001.10 T 0345-001.15 T 0350-001.20
1	LITERAL	S 0376-001.37 S 0398-002.08 S 0425-002.25 S 0433-002.29 S 0434-002.29 S 0442-002.33 S 0443-002.33 S 0450-002.36 S 0451-002.36 S 0475-003.01 S 0496-004.07 S 0499-004.15 S 0507-004.16 S 0522-004.23 S 0570-005.18 S 0572-005.22 S 0578-005.29 T 0586-005.36 T 0592-005.38 T 0596-005.40 T 0602-005.42 T 0606-005.44 S 0626-005.51 S 0632-006.06 T 0633-006.07 S 0638-006.11 T 0644-006.15 T 0652-006.16 T 0654-006.18 T 0655-006.16 T 0674-006.25 T 0676-006.27 T 0679-006.29 T 0679-006.29 T 0681-006.31 T 0689-006.36 T 0693-006.36 T 0698-006.36 T 0698-006.36 T 0702-006.38
2	LITERAL	S 0426-002.25 T 0488-004.01 T 0501-004.12 S 0575-005.28 T 0610-005.46 T 0653-006.16 T 0690-006.36 T 0694-006.36 T 0699-006.36 T 0699-006.36
23	LITERAL	T 0482-003.05
3	LITERAL	S 0468-002.44 T 0628-006.01 T 0673-006.25 T 0691-006.36 T 0695-006.36 T 0700-006.36 T 0700-006.36
4	LITERAL	S 0468-002.43 T 0692-006.36 T 0696-006.36 T 0701-006.36 T 0701-006.36
49	LITERAL	T 0630-006.04 T 0680-006.29
5	LITERAL	T 0375-001.35 S 0375-001.36
50	LITERAL	S 0683-006.32 T 0686-006.32 T 0698-006.36 T 0699-006.36 T 0700-006.36 T 0701-006.36
7	LITERAL	S 0376-001.37
A	UNDEFINED	T 0353-001.22
ACCEPT-DATE-300	0361-001.29	E 0393-002.03
ACCEPT-NUM-TAPES-300	0379-001.40	E 0402-002.11
ACN	UNDEFINED	T 0336-001.03 T 0337-001.05 S 0338-001.07 T 0340-001.08 T 0341-001.10 S 0342-001.12 T 0344-001.13 T 0345-001.15 S 0346-001.16 T 0349-001.18 T 0350-001.20 S 0351-001.21
AMT-ADV-PAY-CTR	0105	R 0449-002.36 S 0592-005.39 S 0593-005.39 R 0616-005.48
AMT-NOR-PAY-CTR	0101	R 0424-002.25 R 0441-002.33 S 0588-005.37 R 0612-005.48
AMT-PAR-PAY-CTR	0103	R 0432-002.29 S 0598-005.41 S 0599-005.41 R 0614-005.48
AMT-UNMAT-PAY-CTR	0093	NOT REFERENCED
B	UNDEFINED	T 0353-001.22
BAD-NUM-TAPES-306	0394-002.04	E 0383-001.41
BATCH-CTRS	0319	NOT REFERENCED
BATCH-DETAIL-1	0305	S 0702-006.38
BATCH-HEAD-1	0283	S 0672-006.25
BATCH-HEAD-2	0289	S 0673-006.25
BATCH-HEAD-3	0297	S 0674-006.25
BATCH-SUB	0111	R 0397-002.08 R 0398-002.08 R 0402-002.10 R 0426-002.25 R 0434-002.29 R 0443-002.33 R 0451-002.36 R 0468-002.43 R 0469-002.44 T 0488-004.01 T 0489-004.02 T 0490-004.03 T 0501-004.12 T 0502-004.13 T 0503-004.16
BATCH-TOT-ADD-400	0481-003.05	P 0465-002.40 P 0479-003.03

SAMPLE DIAGNOSTIC FLOW SUMMARY

```
                                              I
                           0361  0369         V                 01.29
     E 0393 ------->*-----------------------------------------------*
                    *              (ACCEPT-DATE-300)                  *
                    *-----------------------------------------------*------->E 0388
                                              I
                                              I
                           0370  0387         V                 01.33
     E 0402 ------->*-----------------------------------------------*          E 0436 E 0409 E 0420 E 043
                    *               (TEST-GO-DEP)                     *          E 0453 E 0505 E 0509 E 038
                    *           (ACCEPT-NUM-TAPES-300)                *          E 0436 E 0453 E 0495 E 050
                    *      ALTER  0409 TO GO TO  0420                 *          E 0563 E 0521 E 0581 E 052
                    *-----------------------------------------------*------->E 0524 E 0394 E 0403
                                              I

                          0378 RESERVED WORD MARGIN A

                                              I
                           0388  0393         V                 01.47
 E 0372 E 0367 ------->*-----------------------------------------------*
                    *              (INVALID-DATE-305)                 *
                    *-----------------------------------------------*------->E UNDF E 0453 E 0361
                                              I

                          0390 NO LITERAL CLOSING QUOTE
                          0392 UNDEFINED REFERENCE
                          0392 NESTED CONDITIONAL

                                              I
                           0394  0402         V                 02.04
     E 0383 ------->*-----------------------------------------------*
                    *             (BAD-NUM-TAPES-306)                 *
                    *-----------------------------------------------*------->E 0453 E 0436 E 0379
                                              I

                          0396 NON-ANSI CHARACTER SEQ.
                          0397 NESTED CONDITIONAL
                          0399 NESTED CONDITIONAL

                                              I
                           0403  0408         V                 02.12
E 0480 E 0466 E 0387 ------->*-----------------------------------------------*
              X 0554        *           (GET-NEXT-DISK-REC-310)              *          P 0509 - 0519 E 0657 P 05
                    *-----------------------------------------------*------->- 0554 E 0467 E 0467
                                              I
                                              I
                           0409  0419         V                 02.17
A 0378 E 0414 E 0370 ------->*-----------------------------------------------*
                    *             (MATCHED-BILL-320)                  *
                    *      ALTER  0492 TO GO TO  0436                 *
                    *-----------------------------------------------*------->A 0378 - 0420 E 0409
                                              I

                          0410 STATEMENT UNALTERABLE
                          0414 POSSIBLE LOOP
                          0413 NESTED CONDITIONAL
                          0420 PARAGRAPH ENDS BADLY

                                              I
                           0420  0427         V                 02.24
E 0371 A 0378 - 0409 ------->*-----------------------------------------------*
                    *             (CHECK-ZERO-320)                    *
                    *                 (EXIT)                          *
                    *-----------------------------------------------*------->E 0428 E 0453 X 0444
                                              I

                          0427 EXIT AFTER GO

                                              I
                           0428  0435         V                 02.28
     E 0421 ------->*-----------------------------------------------*
                    *             (CHECK-LESS-321)                    *
                    *-----------------------------------------------*------->E 0437 E 0453
                                              I
                                              I
                           0436  0436         V                 02.31
E 0372 A 0420 - 0492 ------->*-----------------------------------------------*
              E 0401        *                (TOTAL-OUT)                     *
                    *  NULL PARAGRAPH                                 *
                    *-----------------------------------------------*
                                              I

                          0437 NO VERB IN PARAGRAPH

                                              I
                           0437  0445         V                 02.32
X 0420 E 0429 E 0371 ------->*-----------------------------------------------*
                    *             (CHECK-EQUAL-322)                   *
                    *-----------------------------------------------*------->E 0446 P 0420 E 0453
                                              I
                                              I
                           0446  0452         V                 02.36
     E 0438 ------->*-----------------------------------------------*
                    *            (IT-IS-GREATER-323)                  *
                    *-----------------------------------------------*------->E 0453
                                              I
                                              I
```

SUPEREF

Mantech Corporation

GENERAL

SUPEREF accepts FORTRAN coded source programs and produces a comprehensive symbolic name cross reference dictionary.

SUPEREF is operational on any CDC 6000 series computer whose software capability has been updated to include the Random access (mass storage) File Routines OPENMS, STINDX, READMS, and WRITMS.

PACKAGE OUTPUT

Variable Name Dictionary - This listing includes the variable name being used, the routine that it is located in, and various other information.

Note: A note of interest is that Mantech Corporation was awarded two contracts from the U. S. Naval Ordinance Laboratory in Silver Spring, Maryland to have two large-scale FORTRAN simulation programs processed by SUPEREF.

```
      PROGRAM INTEST2 (INPUT,OUTPUT,SAVEM,
     1                 TAPE5=INPUT,TAPE6=OUTPUT,TAPE7=SAVEM)
      COMMON NOLN6,NOLN7,LNAL6,LNAL7,NPAGE6,NPAGE7
      COMMON /AHCD/ A(5),B(5,2),C(10),D(10,2,2),F,G(10,4)
      DIMENSION E(106)
      EQUIVALENCE (E(1),A(1))
      DIMENSION ITRTBL(7,2),IDMTBL(6,3),LOADR(3,3),VALUE(3)
      DATA ITRTBL, IDMTBL
     1/1HA,1HB,1HC,1HD,1HF,1HG,6H))))))),
     2  0, 5, 15, 25, 65, 66,     106,
     35,5,10,10,1,10,  0,2,0,2,0,4,  0,0,0,2,0,0  /
      DATA ISW /0/
      NOLN6=0
      NOLN7=0
      LNAL6=60
      LNAL7=60
      NPAGE6=0
      NPAGE7=0
      CALL LINE6 (60)
      CALL LINE7 (60)
5     READ (5,100) NAME
100   FORMAT (A6)
      IF (NAME.EQ.6HENDINP) GO TO 30
      DO 10 I=1,6
      IF (NAME .EQ.ITRTBL(I,1)) GO TO 15
10    CONTINUE
      WRITE (6,200) NAME
200   FORMAT (1H0,A6,26H IS NOT IN TRANSFER TABLE.,/,
     11H0,47HINTEST2 PROGRAM PROCESSING IS BEING TERMINATED. )
      CALL LINE6(2)
      STOP
15    INCRM=ITRTBL(I,2)
20    READ (5,300) (LOADR(J,1),LOADR(J,2),LOADR(J,3),VALUE(J),J=1,3)
300   FORMAT (3(3I4,F10,3))
      DO 25 K=1,3
      IF (LOADR(K,1).EQ.9999) GO TO 5
      IF (LOADR(K,1).EQ.0) GO TO 20
      IF ((LOADR(K,1).LE.IDMTBL(I,1)).AND.
     1    (LOADR(K,2).LE.IDMTBL(I,2)).AND.
     2    (LOADR(K,3).LE.IDMTBL(I,3))        ) GO TO 23
      WRITE (6,360) LOADR(K,1),LOADR(K,2),LOADR(K,3),VALUE(K),NAME,
     1              IDMTBL(I,1),IDMTBL(I,2),IDMTBL(I,3)
360   FORMAT (1H ,8H*INDEX (,I4,1H,,I4,1H,,I4,9H), VALUE ,F10.3,
     120H FOR ARRAY/VARIABLE ,A6,1H(,I4,1H,,I4,1H,,I4,1H) )
      CALL LINE6 (1)
      ISW=1
      GO TO 25
23    II=IDMLOC(IDMTBL(I,1),IDMTBL(I,2),IDMTBL(I,3),
     1          LOADR(K,1),LOADR(K,2),LOADR(K,3)) + INCRM
      E(II)=VALUE(K)
      WRITE (7,350) NAME,LOADR(K,1),LOADR(K,2),LOADR(K,3),VALUE(K)
350   FORMAT (1H ,7X,A6,8X,1H(,I4,1H,,I4,1H,,I4,1H),3X,F10.3)
      CALL LINE7 (1)
25    CONTINUE
      GO TO 20
30    IF (ISW.NE.0) GO TO 35
      WRITE (6,305)
      CALL LINE6 (1)
305   FORMAT (1H0,39HNO DIAGNOSTICS FOR THIS RUN OF ITEST2. )
35    WRITE (6,500) ITRTBL(1),A
      CALL LINE6 (2)
      WRITE (6,500) ITRTBL(2),B
      CALL LINE6(3)
      WRITE (6,500) ITRTBL(3),C
      CALL LINE6 (3)
      WRITE (6,500) ITRTBL(4),D
      CALL LINE6 (9)
      WRITE (6,500) ITRTBL(5),F
      CALL LINE6 (2)
      WRITE (6,500) ITRTBL(6),G
      CALL LINE6 (9)
500   FORMAT (1H ,A6,/,(5(F10.3)))
      ENDFILE 7
      CALL EXIT
      END
```

B-135

```
********************************     THIS IS A -S U P E R E F- LISTING OF     ********************* PAGE    1**

                                     I L L U S T R A T I V E    P R O B L E M

 ROUTINE                                      FORTRAN STATEMENT                                          VARIABLE
  NAME                                                                                                     NAME

 INITL         ***** COMMON / ABCD  /                                               BD*   20......      A
 INITL               DATA A /5*1.0/                                                 BD*   30......      A
 INTEST2       ***** COMMON / ABCD  /                                               IN*   40......      A
 INTEST2             EQUIVALENCE (E(1),A(1))                                        IN*   60......      A
 INTEST2        35   WRITE (6,500) ITRTBL(1),A                                      IN*  600......      A

 INITL         ***** COMMON / ABCD  /                                               BD*   20......      ABCD
 INTEST2       ***** COMMON / ABCD  /                                               IN*   40......      ABCD

 INITL         ***** COMMON / ABCD  /                                               BD*   20......      B
 INITL               DATA B /10*2.0/                                                BD*   40......      B
 INTEST2       ***** COMMON / ABCD  /                                               IN*   40......      B
 INTEST2             WRITE (6,500) ITRTBL(2),B                                      IN*  620......      B

 INITL         ***** COMMON / ABCD  /                                               BD*   20......      C
 INITL               DATA C /10*3.0/                                                BD*   50......      C
 INTEST2       ***** COMMON / ABCD  /                                               IN*   40......      C
 INTEST2             WRITE (6,500) ITRTBL(3),C                                      IN*  640......      C

 INITL         ***** COMMON / ABCD  /                                               BD*   20......      D
 INITL               DATA D /40*4.0/                                                BD*   60......      D
 INTEST2       ***** COMMON / ABCD  /                                               IN*   40......      D
 INTEST2             WRITE (6,500) ITRTBL(4),D                                      IN*  660......      D

 INTEST2             DIMENSION E(106)                                               IN*   50......      E
 INTEST2             EQUIVALENCE (E(1),A(1))                                        IN*   60......      E
 INTEST2             E(II)=VALUE(K)                                                 IN*  500......      E

 INTEST2             CALL EXIT                                                      IN*  740......      EXIT

 INITL         ***** COMMON / ABCD  /                                               BD*   20......      F
 INITL               DATA F /5.0/                                                   BD*   70......      F
 INTEST2       ***** COMMON / ABCD  /                                               IN*   40......      F
 INTEST2             WRITE (6,500) ITRTBL(5),F                                      IN*  680......      F

 INITL         ***** COMMON / ABCD  /                                               BD*   20......      G
 INITL               DATA G /40*6.0/                                                BD*   80......      G
 INTEST2       ***** COMMON / ABCD  /                                               IN*   40......      G
 INTEST2             WRITE (6,500) ITRTBL(6),G                                      IN*  700......      G

 IDMLOC              FUNCTION IDMLOC (II,JJ,KK,I,J,K)                               ID*   10......      IDMLOC
 IDMLOC              IDMLOC=I+II*(J-1)+II*JJ*(K-1)                                  ID*   40......      IDMLOC
 IDMLOC         10   IDMLOC=I+II*(J-1)                                              ID*   60......      IDMLOC
 IDMLOC         20   IDMLOC=I                                                       ID*   80......      IDMLOC
 INTEST2        23   II=IDMLOC(IDMTBL(I,1),IDMTBL(I,2),IDMTBL(I,3),                 IN*  480......      IDMLOC
 INTEST2          1            LOADR(K,1),LOADR(K,2),LOADR(K,3)) + INCRM            IN*  490......      IDMLOC

 INTEST2             DIMENSION ITRTBL(7,2),IDMTBL(6,3),LOADR(3,3),VALUE(3)          IN*   70......      IDMTBL
```

B-135

FORDOC

J. Toellner and Associates

GENERAL

FORDOC accepts Fortran source program as input and produces a restructured source deck along with a variable name cross-reference. The system is composed of six modules that can be run separately or as an entire series.

PACKAGE OUTPUT

Restructured source deck - the original source deck is "cleaned up" by FORDOC giving an easier to read source deck.

Cross-Reference - this listing is a variable name cross-reference listing references made to a given variable name.

SUMMARY

SUMMARY

The need for adequately documented computer programs is an essential element common to all data processing centers. This need for documentation can be efficiently accomplished through the use of automatic documentation software packages. The use of these packages saves both time and money as well as establishing a standardized documentation presentation for each program written.

This report has shown most of the better proprietary automatic documentation systems along with their output features. RPG documentors have been excluded intentionally.

It is evident that there is much room for improvement in this area of program documentation. The door is open for an automatic documentation system which does a better job of presenting a program structure. Better automatically produced flowcharts and descriptions of data structures are two areas in which work needs to be done to develop new modes of presentation.

There also is a total lack of any aids to produce global or system-wide documentation in the software packages reviewed here.

Perhaps through the combinations of certain features found in automatic documentation packages existing today, and by utilizing new ideas to supplement these features, an ideal system could be developed.

APPENDIX C

COMPARISON OF OPERATING SYSTEMS, IBM S/360 OS, UNIVAC EXEC 8, CDC SCOPE 3

AUTOMATIC SYSTEM FOR COMPUTER PROGRAM DOCUMENTATION

Working Paper No. 5

July 14, 1972

Documentation and Operating Systems

by

Andrew Sobey, Jr.

Texas A&M University

Texas Engineering Experiment Station

C-1a

ABSTRACT

The various options available to the user through compilers (PL/I, FORTRAN, and COBOL), assemblers and linkage editors/loaders are listed. Conclusions are then drawn as to which of these options would be useful in the development of an automatic documentation system.

TABLE OF CONTENTS

TABLES

INTRODUCTION

The purpose of this paper is to describe the interaction between a user program and the operating system of a computer. The motivation for this study (also the main objective of the study) was 1.) to determine those options available to the user without modification of the operating system which would be of significant importance to the development of an automatic program documentation system and 2.) to determine what difficulties might be encountered at the operating system level in developing an automatic documentation system to be used on many different manufacturers' computers. As the basis for the discussion of this topic, the following three operating systems will be considered:

1. IBM's OS/360
2. UNIVAC's EXEC 8
3. CDC's SCOPE 3

To explain the medium by which this interaction takes place, and also the extent to which it can take place, the compiler and linkage editor/loader options available to the user through the control language for these systems will be discussed. As it would be impossible in the allotted time to examine all existing higher level languages and their compiler options (just as it would also be impossible to study all operating systems in existence), the following three languages have been chosen for study as being representative of all higher level languages: PL/I, FORTRAN, and COBOL. Also, the assembler languages of the above mentioned three machines and their options will be considered.

COMPILER/ASSEMBLER OPTIONS

A. Options

The first step in the study was to compile a list of all the compiler/assembler options available to the user by the operating systems for the four languages under consideration. These options were then studied in light of their possible usefulness as part of an automatic system of program documentation. A list of these options is indicated on the next several pages. It will contain an explanation of the option, and is broken down by manufacturer and language.

It is urged that the reader study this list of options before proceding to the next section of the paper (See list on following pages).

B. Results and Remarks

Because one of the criteria of the study was to determine those items which could be obtained without modification to the operating system, let us consider those options which are common to FORTRAN, COBOL, and Assembler languages under all operating systems (PL/I is not offered on the UNIVAC 1108 or CDC 6600 machines.)

1. A source listing may be obtained.
2. A cross-reference table may be obtained.
3. The user may specify from where to read the input file.
4. The user may specify where to write the object module.

Thus, it can be seen that although a myriad of options are offered by the three computer manufacturers, only four options are common to all of them. These options, however, the author feels would be of significant value to the development of an automatic documentation system.

TABLE I. ASSEMBLER OPTIONS

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Specify that an assembler language listing is to be produced	L=fn; place a full listing on the file named fn; if L=0, put a brief listing on the file named OUTPUT; if absent OUTPUT is assumed for fn	LIST (NO)	
Specify from what file the input is to be read.	I = fn; the input is on the file named fn; if absent INPUT is assumed for fn.	Done through JCL	The file name to be read is coded on the control statement.
Specify the file the object module is to be written upon.	B=fn; a binary file is written on the file named fn; if absent, LGO is assumed for fn; if B=0 is coded, the binary file is suppressed.	LOAD, pace on the drive specified by the SYSGO DD statement	The file name the object module is to be placed on is specified on the control statement
Specify where to search to find the systems text.	if absent, or S, the systems text is on SYSTEXT(SCOPE Central Processor Macros) S=rname; the systems text is on the overlay named rname. S=SCPTEXT; the systems text if from the library overlay named SCPTEXT which contains the system symbol definitions S=SMTEXT; systems text for SORT/MERGE macros plus SCPTEX macros	Done through JCL	

OPTION	CDC 6600	IBM 360	UNIVAC 1108
	S=PPTEXT; systems text for PP macros (SCOPE Peripheral Processor Macros)		
Specify that a deck is to be produced of the object module		DECK(NO)	P
Specify that the object module is to be tested.		TEST(NO); the object module contains the special source symbol table required by the test translator (TESTRAN) routine and the TSO command processor	
Specify that a cross-reference table is to be produced.	L	XREF (NO)	L
Specify that the assembler check for possible coding violations of program reenterability		RENT (NO)	
Specify the number of lines to be printed between headings in the listing.		LINECNT=xx; where xx is the number of lines desired.	
Specify that boundary alignment errors are to be printed		ALGN (NO)	

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Specify what capability the assembler has.		OS-complete OS assembler F capability DOS-DOS assembler D and F capability	
Specify an optional file upon which to write diagnostic messages		TERM(NO); write the diagnostic messages on the SYSTERM data set	
Specify numbers are to be written in the beginning of each line for which diagnostic information is given		NUM(NO); the line number field (cols 73-80) or TSO through the EDIT command supply these numbers; this option is only valid with TERM above	
Specify that statement numbers are to be supplied for statements for which diagnostic information is given		STMT(NO); these numbers will be written on the SYSTERM data set; this option is only valid with 'TERM' above	
Specify that the run is to be continued even though errors have been detected		Done through JCL	A; this option need not be specified as it is always in effect unless overridden by the X option
Specify that the run is to take the error exit if errors are detected		Done through JCL	X; overrides the A option above

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Indicate the relocatable is output code is quarter-word sensitive.			F
Specify compressed card input in columns 1 through 80			G
Specify that input cards have sequence numbers in columns 73-80			H; these numbers are ignored unless the K option (below) is specified
Specify compressed card input in columns 1-72 and sequence numbers in column 73-80.			J
Specify a sequence check is to be performed when reading the input cards (columns 73-80)			K; used in conjunction with the H or J option above
Specify the assembler's internal symbol table and procedure sample table areas are to be expanded			M; expanded by an additional 10240 words when used with the R option these areas are expanded by only 5120 words
Specify all listings are to be suppressed		NOLIST	N; or omit all listing parameters
Specify a machine language listing is to be produced			O; done in octal

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Specify the source language output should be in Fieldata code			P
Specify the ASCII character set is to be used		Done through JCL	Q
Specify that the amount of core storage is to be lessened for the internal tables			R; 5120 words are dropped from the assemblers: internal symbol and procedure sample table
Specify that the relocable output code is third-word sensitive			T; (is overridden by the F option)
Specify an update is to be made of an existing source language input element to the next higher element cycle			U; the 'eltnames' parameter on the control statement must be coded; the updated version has the same name as before but the cycle number is increased by one
Specify the correction lines are to be listed at the head of the printer listing			W
Specify the name and version of the source element		Done through JCL	N1/V1; where N1 is the element name and V1 in the version number

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Specify the name and version of an updated source language module, ie, where to place the updated module		Done through JCL	N2/V2; where N2 is the element name and V2 is the version number
Specify the name and version of the updated object module, ie, where to store it		Done through JCL	N3/V3; where N3 is the element name and V3 is the version number
Specify that corrections are to be noted on the listing			C
Specify alternate Assembler is to be used			J
Specify a set of system symbol definitions are to be provided to the assembler before the assembler source code			M
Specify comments are to be inserted in the source or object module			K
Specify no listing is desired			N
Specify the output punched is in multiple word octal format			Q
Specify an updated output source language element is to be punched on cards			S
Specify the correction deck is to be listed prior to the assembler listing			W

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Suppress formation of information normally given to the diagnostic routine			Z

TABLE II. FORTRAN OPTIONS

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Specify that a machine language listing of the compiled program is to be produced	O; in octal		
Specify that a source listing of the program to be compiled is to be produced	L	SOURCE(NO) EDIT(NO); produces structured listing	I; single spaced listing L; detailed listing M; double spaced listing N; produce no listing
Cross Reference Table-Showing where the variables were defined and where they were referenced	R	XREF(NO)	L
List All diagnostics indicating all non-ASA usage	X		
Specify an assembler-language listing is to be produced		LIST(NO)	L
Tells on which file to write the output	l=fn; l is the type of listing desired (one of the above four); fn is the file name which is to be written upon	Done through JCL	Done through control statement
Indicate from what file the source program is to be read	I=fn; fn tells which file is to be read from	Done through JCL	Done through the Control Statement

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Indicate where to write the machine language file of the compiled program	B=fn; a binary relocatable file is to be written on the file-name fn E-the object code is prepared for EDITSYM-this facilitates hand optimization of the compiled code	LOAD; write the object module on the data set specified by the SYSLIN DD statement	Done through the control statement
Provide and error trace back and calling sequence	T	ID; also, internal statement numbers are generated for statements that call a subroutine or contain an exteranl function references (four extra bytes are needed for linkage)	Done by 'SNOOPY'
permits the programmer to name his main program routine		NAME = xxxxxx; where xxxxxx is a six character name conforming to FORTRAN rules of variable names	
Indicate the maximum number of lines to be written on a page of the source listing		LINECNT=XX; where xx is the number of lines desired	
Indicate an object module is to be punched		DECK(NO)	P
Include a table of named variables, their type and location, and a table of labels		MAP(NO)	

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Indicate which character set the compiler should accept		BCD or EBCDIC	
Indicate to what degree the compiler is to optimize the coding produced		OPT=0; no optimization OPT=1; each source module is treated as a single program loop each loop is optimized with regard to register allocation and branching OPT=2; the compiler treats each source module as a collection of program loops and optimize each loop with regard to register allocation, branching,common expression, elimination, and replacement of redundant computations	
Indicate the amount of main storage available to the compiler		Size=nnnnk; where nnnn is between 115 and 9999	
Specify that a listing indicating the loop structure and the logical continuity of the program is to be produced		EDIT; A SYSPRINT DD statement must be included, and OPT=2 must also be specified	

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Specify whether or not the output from the compiler is to be accepted if an error has been detected		Done through JCL	A; accept it X; abort the run
Specify inline double precision coding is to be generated			D
Specify the labeled common is to be attached to only those segments which use it			G
Specify the alternate FORTRAN compiler is to be used			J
Specify an updated source module is to be punched			S
Specify the time of each phase and the total compilation time is to be printed		not a user option; specified when the operating system is generated	T
Specify that the correction deck is to be listed prior to the compilation listing			W
Specify patch cards are to be used to alter the compiler operation.			Y; is suggested that only systems programmers use this option
Specify that the formation of information normally given to the diagnostic-routine is to be suppressed			Z

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Specify what source element is to be updated		Done through JCL	N1/V1; when N1 is the source element name and V2 is the version number
Specify an updated source element is to be stored		Done through JCL	N2/V2; where N2 is the source element name and V2 is the version number
Specify where to store a relocatable object module		Done through JCL	N3/V3; where N3 is the source element name and V3 is the version number

TABLE III. COBOL OPTIONS

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Indicate that the source module is to be listed	L	SOURCE(NO)	I; single spaced listing L; includes items specified by C,D and O options
List all diagnostics indicating non-ASA usage	X		
Indicate that items copies from the library are to be listed	C		
Indicate that a cross reference table is to be produced	R; lists cross reference pointers to source lines	XREF; lists data names and procedure names	R; lists data names and file names
Indicate that the object module is to be listed	O; listed in octal	PMAP; listed in hexidecimal also lists register assignments, global tables, and literal pools	O; listed in octal
Indicate where to write the file containing the last output	l=fn; where l is L or any combination of L with X,C,R,O,M (see above) and fn is the file name on which the output is to be written	Done through JCL	

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Indicate from what file the source program is to be read	I=fn; where fn is the filename to be read	Done through JCL	Done through the control statement
Indicate on what file to write the object module	B=fn; write the relocatable binary file named fn; B=0 suppresses the writing of this file	LOAD; place the object module on mass storage or a tape volume	Done through Control Statement
Indicate on what file the COBOL library is located	S=fn; fn is the file on which the COBOL library is located; needed only if fn is not COLIB	Done through JCL	
Suppress all DATA DIVISION binary output except from the WORKING STORAGE SECTION and CONSTANT SECTION for a subcompiled program which would duplicate output from a separately combined main program-this enables them to be properly loaded together	SUB		
Separate the overlay segments from the main programs so that separately compiled programs can be loaded properly	OB=fn; the overlay segments are written on the file named fn		
Specify the amount of main storage available for compilation		SIZE=yyyyyyy; is measured in bytes	
Specify the amount of storage allocated for buffers		BUF=yyyyyy. measured in bytes; if BUF and SIZE are both specified,	

OPTION	CDC 6600	IBM 360	UNIVAC 1108
		the amount of storage in BUF is included in SIZE	
Specify that a condensed listing is to be produced		CLIST(NO): the procedure portion of the listing will contain generated card numbers, verb references and the location of the first generated instruction for each verb	
List a glossary of symbols used in the program		DMAP(NO)	
Specify an object module deck is to be produced		DECK(NO)	P S; a deck of an updated file is produced
Provide a sequence check of the source module statements		SEQ(NO)	
Indicate the number of lines to be printed on each page of the source listing		LINECNT=xx; where xx is the number of lines desired	
Indicate the severity of error messages to be printed		FLAGN; list all warning and diagnostic messages FLAGE; list all but the warning messages	

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Indicate the object code listing, object module and link edit decks are to be suppressed if an E level error message is generated by the compiler		SUPMAP(NO)	
Indicate the type of spacing desired on the source listing		SPACE1; single space SPACE2; double space SPACE3; triple space	
Indicate what kind of movement of computational fields is desired		TRUNC; if the number of digits in the sending field is greater than the number of digits in the receiving field, the arithmetic item is truncated to the number of digits specified in the PICTURE clause of the receiving field NOTRUNC;the movement of the item depends on the size of the receiving field (fullword, halfword)	
Indicate which character should be accepted to delineate literals and to be used in the generation of figurative constants		QUOTE; (") is the acceptable character APOST; (') is the acceptable character	
Specify the output from the compiler is to be accepted even if errors are detected		Done through JCL	A

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Specify that the checking of punched card sequence numbers (columns1-6) is to be ignored			B
Specify the matched names of CORRESPONDING data names are to be listed			C
Specify an alphabetic list of all data names and file names is to be printed			D
Specify a detailed list of diagnostics is to be printed			E
Specify the alternate COBOL compiler is to be called			J
Specify that a list of items requested through the COPY and INCLUDE verbs is to be produced			K
Specify a list of procedure - names with the same first five characters is to be produced			M
Specify that the source listing is to contain only those items requested by options			N; if absent E, I, and K are assumed
Specify the updated module is to be punched on cards			S

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Specify that the label will consist of the first five characters preceded by one element - specified unique character			T
Specify that the contents of columns 72-80 are to be ignored			U
Specify that the compiler has encountered a subprogram rather than a main program			V; the generation of a starting address is suppressed
Specify that the correction deck is to be listed before the compilation listing			W
Specify the run is to be aborted if an error is found		Done through JCL	X
Specify that the formation of information normally given to the diagnostic routine is to be suppressed			Z
Specify the name and version of the source element		Done through JCL	N1/V1; where N1 is the name of the source element and V1 is the version number
Specify the name and version of an updated source language module, ie. where to store it		Done through JCL	N2/V2; where N2 is the name of the module and V2 is the version number

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Specify the name and version of the updated object module, ie, where to store it		Done through JCL	N3/V3 where N3 is the object module name and V3 is the version number

TABLE IV PL/I OPTIONS

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Specify the amount of main storage available for compilation		SIZE=yyyyyy; indicates yyyyyy bytes of storage are available SIZE=yyyK; indicates yyyK bytes of storage are available SIZE=999999; instructs the compiler to obtain as much main storage as it can	
Specify the type of compiler optimization desired		OPT=0; keep object program storage requirements to a minimum at the expense of execution time OPT=1; causes object program execution time to be reduced at the expense of storage OPT=2;includes OPT=1, but also requests the compiler to optimize the machine instructions generated for certain DO-loops and expressions in subscript lines	
Request the compiler to produce additional instructions that will allow statement numbers from the source program to be included in diagnostic messages produced during execution of the compiled program		STMT(NO)	

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Permit the programmer to name the load module that will be created by the linkage editor		OBJNM=xxxxxxxx; must begin with an alphabetic character	
Permit the operating system to properly handle interrupts on the IBM models 91 and 195		OBJIN; if execution takes place on the IBM 91 or 195 computers OBJOUT; if execution does not take place on the IBM 91 or 195 computers	
Cause the compiler to construct larger dictionaries		EXTDIC; of the dictionary block size is 1K bytes a dictionary 1.5 times that of normal size is used; the dictionary used is 3.5 times normal size if the block size was greater than 1K bytes	
Specify the condition for termination after syntax checking if errors are detected		SYNCHKE; if errors of severity "ERROR" or above are found SYNCHKS; if errors of severity 'SEVERE' or above are found SYNCHKT; if errors of severity 'TERMINATION' are found	
Specify that the preprocessor is to be used		MACRO(NO)	

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Specify the compiler is to immediately compile the source module produced by the preprocessor		COMP(NO)	
Specify the output from the preprocessor is wanted in the form of a card deck		MACDCK(NO)	
Specify whether the source statements are written in the 48 or 60 character set		CHAR60; written in the 60 character set CHAR48; written in the 48 character set (the compiler will accept both if CHAR48 is coded, however)	
Specify whether the EBCDIC or BCD character sets is to be accepted		EBCDIC; BCD	
Specify the extent of the part of each input record that contains PL/I source statements. It can also specify the position of the ANS carriage control character desired to control the format of the listing produced		SORMGIN=(mmm,nnn[,ccc]) where: mmm is the first byte of the field that contains the source statements nnn is the last byte of the source statement field ccc is the byte containing the carriage control character and mmm < nnn <100 and ccc < mmm or ccc > nnn	

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Specify where to place the object module produced		LOAD; place the object module on the SYSLIN data set	
Specify that a card deck is to be produced of the object module		DECK; columns 73-76 contain a code to identify the object module and columns 77-80 contain a 4 digit serial number	
Specify the number of lines to be printed on a page of the source listing		LINECNT=XX; where xx is the number of lines desired	
Specify that the compiler options are to be listed at the start of compilation		OPLIST(NO)	
Specify that a listing of the source statements input to the preprocessor is to be produced		SOURCE2(NO)	
Specify that a listing of the source statements processed by the compiler is to be produced		SOURCE(NO); the statements are either the original source statement or the output from the preprocessor	
Specify that for each statement of a DO-group, the block and nesting level should be indicated on the source listing		NEST(NO)	

OPTION	CDC 6600	IBM 360	UNIVAC 1108
Specify that a table of source program identifiers and their attributes should be produced		ATR(NO); also an aggregate length table, giving the length in bytes of all major structured an non-structured arrays will be produced	
Specify that a cross-reference table is to be produced		XREF(NO); if ATR(above) and XREF are both specified, the table is combined	
Specify that the source listing should contain the ESD		EXTREF(NO)	
Specify that a list of the machine instructions generated by the compiler is to be produced		LIST(NO); the list is similar to SYSTEM/360 assembler language instructions	
Specify the minimum level of severity that requires a diagnostic message to be printed		FLAGW; list all diagnostic messages FLAGE; list all but warning messages FLAGS; list only 'severe' and 'termination' messages	
Specify that a formatted listing of the compiler modules, compiler storage, and compiler control blocks is to be produced if an unrecoverable error is encountered		DUMP	

The author would now like to list those options provided by any/all of the systems which he feels would be of use to the development of an automatic documentation system.

1. A source listing
2. An object listing
3. A cross-reference table
4. Specify the input source/object file
5. Specify the output source/object file
6. Permit the programmer to name his main program module
7. A table of named variables including their type and location
8. A table of label names
9. An indication of what character set is to be accepted (BCD, EBCDIC, ASCII, FIELDATA, etc.)
10. A listing indicating the loop structure and the logical continuity of the program
11. Specify if an alternate version of a compiler is to be used
12. An indication of the time of compilation, execution, etc.
13. An updated source/object was produced
14. An indication of all non-ASA standard usage
15. An indication of those modules which were subcompiled
16. An indication of the main storage and buffer size
17. An indication of which segments are overlays, which segments they can overlay, and the "path" of the overlay
18. Specify what kind of computational field movement was used
19. Indicate whether the quote (") or the apostrophe (') is to be accepted

20. Indicate what level/capability of compiler/assembler is needed
21. Specify that internal table space have been increased/decreased
22. Specify if the preprocessor was used and a listing of the generated code
23. Indicate what portion of the card contained the source statements

LINKAGE EDITOR/LOADER OPTIONS

A. Options

The second step of the study was to compile a list of the linkage editor/loader options available to the user. (The author regrets that at this time insufficient information could be obtained for the CDC 6600 and UNIVAC 1108 loaders.) These options were then studied in light of their possible usefulness as part of an automatic system of program documentation. A list of the IBM linkage editor/loader options follows.

Linkage Editor Options

1. Downward Compatible - makes the load module processable by either the level E or level F linkage editor - it lets the level E reprocess load modules produced by the level F linkage editor (some differences between the levels E and F linkage editors are: the size of main storage used, the number of entries permitted in the ESD or the RLD, the number of segments allowed, the maximum blocking factor allowed).
2. Hieararchy Format - the programmer can specify his program to be loaded into either processor storage (hieararchy 0) or IBM 2361 core storage (hieararchy 1) - the program can be block or scatter loaded in either area.

3. Not Editable - a module with this attribute has no ESD and can't be reprocessed by the linkage editor. Because no ESD is produced, less space is needed. It is used primarily by the control program.
4. Only Loadable - modules with this attribute can only be brought into core using the LOAD macro - this is done because some subsets of the control program use a smaller control table when a load module is invoked with the LOAD macro, thereby requiring a smaller storage area. It must be entered by a CALL or branch instruction - this attribute is used mainly by the control program and it is suggested that the programmer not use this option because it can impair the usability of a module.
5. Overlay Attribute - is structured according to OVERLAY control statements - overlay modules can only be blocked-loaded, are not refreshable, are not reenterable, are not serially reusable, and cannot be assigned a storage hierarchy.
6. Reusability - this means that the same copy of a load module can be used by more than one task, either concurrently or one at a time.
   a. re-enterable - can be executed by more than one task at a time - this module can't be modified by itself or any other coding during execution.
   b. serially reusable - can be executed by only one task at a time - this type of module must initialize itself or restore any instructions modified during previous execution.
7. Refreshable - can be replaced by a new copy during execution by a recovery rearrangement routine without changing either the sequence or results of processing.

8. Scatter Format - does not need to be loaded in a contiguous block of storage - the programmer specifies the dynamic loading of control sections into noncontiguous or scattered areas within his assigned main storage. The control program can determine a scatter format but for best results the programmer should specify his own control sections which are to be scatter loaded.
9. Test - a module is to be tested and has the symbol tables needed by the test translator (TESTRAN) or the TSO TEST command processor.
10. Exclusive Call - the link editor marks the output module as executable when valid exclusive references have been made between segments (However, a warning message is also issued) - the OVERLAY option must also be specified.
11. LET - when specified, the linkage editor marks a module as executable even though it has found a severity 2 error condition - (example: unresolved external references, valid or invalid exclusive calls requested by an overlay program, an error on a link-edit control statement, a library module can't be found, or there is no more space in the directory of the output module library).
12. No Automatic Library Call - the linkage editor does not call library members to resolve external references, and the output module is still marked as executable (provided no other errors exist)
13. SIZE - Specifies the amount of storage to be used by the level F linkage editor - this option also can be used to specify how much storage is to be used as the load module/text buffer(used to house input/output data).
14. DCBS - lets the user specify the block size for the SYSLMOD data set

in the DCB of a DD statement. SYSLMOD is a DD statement which describes an output module library. It is a partitioned data set on a DASD and has a member name.

15. LIST - lists the control statements processed by the linkage editor on the diagnostic output data set.
16. Module Map - produces a module map on the diagnostic output data set.
17. Cross-reference table - produces a cross-reference table on the diagnostic output data set.
18. Alternate Output or SYSTERM Option - produces the linkage editor error/ warning messages on the SYSTERM data set.

Loader Options

1. MAP - same as 16 above.
2. RES - an automatic search of the link pack area queue is to be made.
3. CALL - an automatic search of the SYSLIB data set will be made.
4. LET - same as 11 above.
5. SIZE - same as 13 above.
6. EP = name - specifies the external name assigned to the entry point of the loaded program.
7. NAME = name - specifies the name to be used to identify the loaded program to the system.
8. PRINT - write informational and diagnostic messages on the SYSLOUT data set.

B. Results and Remarks

All of the above mentioned options would provide useful information to the user. Thus, the author feels that they should all be included in an automatic documentation system.

CONCLUSION

In conclusion, we can see that even though the three general operating systems (IBM, CDC, Univac) reviewed in this report offered a myriad of options, only a few of them are common. However, this does not seem to be a hindrance because the individual compilers, assemblers and linkage-editors/loaders build tables and produce outputs which provide much valuable information for automatic documentation.

At the present time, the operating systems have not been studied to a sufficient depth to determine if any additional information for documentation can be derived from other operating system components. It is hoped that further investigation will also reveal if modification to the operating systems will be necessary to obtain this data.

APPENDIX D

AUDIO DOCUMENTATION EXPERIMENT

A TEST OF AN AUDIO DOCUMENTATION TECHNIQUE

E. R. Story
Roger W. Elliott

Texas A&M University
College Station, Texas
77843

December 1972

D-1-a

ABSTRACT

A technique for documenting programs by means of an audio recording is proposed.

An experiment designed to test the effectiveness of the technique was conducted. The results of this experiment indicate that in some circumstances programmers can use audio documentation as effectively as more conventional forms of documentation.

In general, audio documentation is far easier and cheaper to prepare.

INTRODUCTION

Even though the importance and value of proper documentation have been extensively discussed, most computer programs are not adequately documented. Since the value of documentation is virtually unquestioned and its utilization is so inconsistent, then perhaps an alternative to the conventional documentation package needs to be developed. The purpose of this paper is to present an alternative and to evaluate its effectiveness.

The technique proposed makes use of audio tapes for storing the majority of the documentation for a specific program. An experiment was designed to test the utility of this for program maintenance as compared to more conventional forms of documentation.

A STANDARD FOR AUDIO DOCUMENTATION

The audio documentation package developed for this experiment contains the following four items:

(1) Cover Sheet

(2) Audio Tape

(3) File Information Sheet (if required)

(4) Program Listing.

The cover sheet (fig. 1, p. 3) is a table of contents to the audio tape. The first section provides general information usch as program name, programmer, source language, date, tape speed and type of tape. The second section provides the table of contents for the tape. One of the major disadvantages of the audio technique is the difficulty in finding the desired information. The table of contents is designed to alleviate this problem. For the major components of the program, the table of contents provides: line number of the program listing, paragraph or routine name, and a footage meter reading indicating where on the tape the detailed comments for the particular section may be found. If the programmer desires to listen to a particular portion of the tape, he can then use either the fast forward or fast rewind to position the tape to the desired location.

As can be seen from the table of contents, the tape is composed of two main sections or parts. The first should always contain the following sections: (1) Identification, (2) Overview and I/O, (3) General Flow, and (4) Symbols and Variables. The second major section contains

the detailed descriptions of small segments of the program.

The Identification section always begins by stating the program name. This requirement is imposed for two reasons: (1) To insure the tape has not been mislabeled, and (2) To allow meter synchronization.

The Overview and I/O section includes such items as source language, system or application, entry and exit points, external references, tables, description of input and output, special register usage, etc.

The General Flow section could be considered a verbal flowchart depicting the gross logic of the program. This section describes the major segments of the program and provides the general purpose or function of each major program paragraph or subroutine and indicates the order in which segments are performed or called.

The Symbols and Variables section is self explanatory. Sequentially following the program listing each symbol or variable is named, its line number given and its purpose or usage described.

The most important advantage of audio documentation is provided by the remainder of the tape wherein a detailed description is provided for each paragraph or subsection of the program. The advantage is that much more detail can be presented on the tape than would normally be provided by any of the more conventional forms of documentation.

AUDIO DOCUMENTATION UNIT

Program Name

Revision

0 1 2 3 4 5

Date
Yr Mo Day

Programmer

Phone

___/___-____

Address

Tape Speed

1-7/8 3-3/4 7-1/2

Total Length

________MIN

Tape Format

3" REEL 5" REEL CASSETTE

Table of Contents

Meter Reading

1. Program Identification
2. Program Overview and Input/Output
3. Segmentation and General Flow
4. Glossary
5. Description of Program Segment Logic

a. ____________________________

b. ____________________________

c. ____________________________

d. ____________________________

e. ____________________________

f. ____________________________

g. ____________________________

h. ____________________________

i. ____________________________

j. ____________________________

k. ____________________________

Figure 1 - Audio Documentation Cover Sheet

AN EXPERIMENT

As previously stated, the purpose of this experiment was to test the utility of this form of documentation program maintenance compared to those of the more conventional forms of documentation. One of the first decisions that had to be made was to determine which programming language should be used for the experiment. Since the purpose of the experiment was to evaluate two different types of program documentation, most of the higher level languages were elminated because, to a certain extent, they are all self-documenting. Assembly language was therefore chosen.

The next decision was to limit the programs to between 250 to 350 lines of code. This was made for several reasons, the most important being that a more difficult program would have placed an unreasonable burden on the volunteer test subjects.

Another decision was to determine how well the two programs should be "commented." Any well-developed set of standards for a program documentation package would require that the program listing be well commented, especially if it were written in assembly language. However, a decision was made to include no comments within the source code. This decision was made for two major reasons. The first was that the experiment was being designed to evaluate program maintenance time of audio documentation against the more conventional forms of documentation (mainly flowcharts), and any comments contained in the source listing itself could bias the results. The second reason was

more of a qualitative one. If the two programs were to be partially "commented," then the questions arose as to which parts should be "commented" and the level of detail.

A summary of how the experiment was conducted is presented below.

(1) Two programs were selected. The programs were S/360 assembler programs about 300 statements in length.

(2) For each program an audio documentation package was prepared and a more conventional documentation package consisting of a textual description, flow charts, etc., was also prepared.

(3) 12 Programmers were selected and randomly divided into two groups, 1 and 2.

(4) Programming group 1 was given the audio documentation package for program A, and the conventional documentation package for program B, Group 2 was given the conventional package for program A, and the audio documentation package for program B.

(5) Each programmer was asked to work independently and, using only the documentation provided, make a specified change to each program. The programming time required to make the change using audio documentation and the time required to make the change using conventional documentation were then determined.

(6) An Aspin-Welch t-test analysis was then performed to determine if there were significant differences between the

times.

The experiment's original scope was quite narrow in that it was limited to comparing only the average time required to make changes to a program using audio documentation and the average time required to make the same changes using standard documentation.

No attempt was made to compare the total amount of time required to prepare the respective types of documentation, although the audio documentation is easier to prepare and takes much less time to produce. No attempt was made to perform any type of qualitative evalaation of whether the documentation did meet the established guidelines.

ANALYSIS

When the programmers had completed their changes and the totals calculated as shown on the following page, it was readily apparent that the results were either contradictory or inconclusive. For program "A", the average time of the programmers who had used conventional documentation was less, but for program "B", the average time of the programmers who had used audio documentation was the shorter. When the two programs were compared, "B" had the lower average time, and when totals were run by tapes of documentation, the audio packages had the lower average time.

Thus, it was decided to make four tests. The first two tested the average time of audio documentation versus conventional documentation for the two programs separately. For these, a two-tail test at the 0.05 level of significance, was conducted. The third test compared the two programs themselves, i.e. the average time required to change program "A" vs. the average time to change program "B". This was again a two-tail test; however, it was at the 0.1 level of significance. A larger than usual risk of making a Type I error could be accepted for this test in order to reduce the risk of making a Type II error. This risk needed to be reduced because test four should not have been conducted unless the hypothesis of test three was accepted. Test four compared the overall average time to make the changes to both programs using audio documentation vs. the overall average time

Times Required to Make Changes

Programmer	Program "A" Audio	Program "A" Conventional	Program "B" Audio	Program "B" Conventional
1	59			72
2		43	34	
3		35	20	
4	39			55
5	67			34
6		42	40	
7		39	38	
8	43			37
9	54			68
10		42	29	
11		35	36	
12	33			22
TOTALS:	295	236	197	288
	531		485	

Total "Audio" Time: 295 + 197 = 492

Total "Conventional" Time: 236 + 288 = 524

Table 1. Times (in minutes) required to make changes to two programs 12 programmers using different types of documentation

using conventional documentation. Obviously, the validity of this test could be subjected to severe criticism if test three indicated a significant amount of difference between the two programs. The fourth test was two-tail test at the 0.05 level of significance. None of the tests turned out to be statistically significant.

OBSERVATIONS & CONCLUSIONS

There are probably several reasons why none of the tests were statistically significant. Some of these are as follows:

1) The programs were too short.
2) The required changes were too minor.
3) An insufficient number of samples was taken.
4) Variances among individual programmers' abilities are so large that only very large differences between sample means will be statistically significant.
5) There are too many variables to isolate whether or not the differences can actually be attributed to the type of documentation used.
6) There actually is no real significance between the sample populations.

The author believed that (1) and (2) above may have played very important roles in this experiment. These limitations were accepted so as not to impose a hardship upon the volunteers who were making the changes. However, these limitations may have precluded the observations of any significant differences in the experiment.

Number (3) above does not appear to be near as critical as (4). The standard deviation among the programmers was so large, that had the difference between the means remained constant, the sample size could have been increased to 120 and still not have been statistically significant.

The large number of variables, number (5) above, is a problem that is extremely difficult to resolve in an experiment of this nature. The ideal situation would be to have only one program that is being changed, by two groups of programmers. One group using audio documentation, the other using conventional. However, this presents the problem of the difference in abilities of the two groups of programmers. Thus, the question could arise as to whether or not the difference should be attributed to the type of documentation used or to programmer abilities. To resolve this question, one can use two programs, and thus both groups get to use both types of documentation. This immediately leads to the comparison of the difficulty of the two programs. All of which brings back the original premise that there may possibly be too many variables involved to obtain a test that is statistically significant. The authors do not believe this is the case, but it is being presented as an alternative possibility.

The last possibility (#6 above) is that there is no significant difference between program maintenance using audio documentation and program maintenance using conventional documentation. Looking back at the data, and after evaluating the four tests, this has to be considered a very strong possiblity.

SUGGESTIONS FOR FURTHER RESEARCH

Although the results of this experiment failed to prove that one type of documentation is better than the other for program maintenance purposes, the authors are convinced that additional research should be conducted into the use of audio techniques. The experiment demonstrated that much more detail can be easily provided by audio documentation than by conventional documentation; however, the experiment as conducted was somewhat limited and may have partially nullified this advantage.

To test fully the effectiveness of the two techniques, longer programs should be used and more complicated changes attempted. The changes should affect more than one section of the programs being tested, and thus, a better evaluation of the differences of the techniques should be provided.

Indications are that audio documentation is easier and faster to prepare; however, a detailed cost analysis study (which was beyond the scope of this project) would be required to prove or disprove this point.

Several of the programmers admitted that they had no real idea as to what to expect from the audio documentation, and that they could probably utilize it better if they were to use it again. In future experiments, programmers should be provided better training in the use of audio documentation prior to the actual experiment itself.

SUMMARY

The experiment conducted in this paper demonstrated that for assembly language programs of approximately 300 lines of code, program maintenance time is approximately the same when using either audio documentation techniques or conventional documentation techniques, thus, audio techniques should be considered as an alternative to the conventional forms of program documentation. Further research should be conducted to determine the effectiveness of the technique for longer programs requiring more complicated changes.

The most important advantage of audio documentation is the volume and degree of detail it can provide about the program. There is absolutely no realistic way that conventional documentation can compete in this area. This should prove to be especially advantageous when the maintenance programmer has not written the original code.

A second advantage of audio documentation may be that it is easier to prepare. Drawing a neat, detailed flowchart using a template is a time consuming job. Drafting and processing the textual material to support a conventional documentation package is even more time consuming.

The detail provided by audio documentation becomes even more important whenever a program listing is poorly commented. Programmers are often hesitant to fully comment a program before it is completely debugged because of the numerous changes that often have to be made. If comments have already been included, then a double change is usually

necessitated. They usually have good intentions of fully commenting the program after it is completely debugged, but for one reason or another, they seldom do. The use of audio documentation greatly alleviates this problem in that the comments are merely placed on the tape.

Perhaps a fourth advantage of the audio documentation technique is that it virtually forces an organized approach to program maintenance, whereas a programmer using conventional documentation may or may not have an organized plan of attack.

Two of the disadvantages of audio documentation are inherent to any sequentially accessed file: (1) It is more difficult to change, and (2) Advance/rewind time to the desired position on the tape may be significant.

Thus, audio documentation is presented as an alternative to the conventional documentation process. Whether or not it should be implemented at any particular data processing center depends upon many factors. These include the characteristics of the projects being documented, the technical environment, and management's commitment to the concept.

APPENDIX E

TEXT EDITORS

AUTOMATIC SYSTEM FOR COMPUTER PROGRAM DOCUMENTATION

Working Paper No. 2

July 14, 1972

Text Editors

by

Hank Goggan

Texas A&M University

Texas Engineering Experiment Station

E-1-a

ABSTRACT

This paper is a summary of current methods available for the editing of text on a computer. A brief discussion of the advantages of using computer-aided text editing for documentation of computer software systems is followed by short reports on present systems, and a conclusion. The principal topic of each report is a contrast of features characteristic of each system.

TABLE OF CONTENTS

INTRODUCTION

"With the advent of inexpensive terminals that communicate directly with a general-purpose computer, there has been a noticeable movement in the computing industry towards utilizing the resources of the computer in many new, non-numerical applications. For example, on-line creation and modification of programs and their documentation have become widely accepted as productive and cost-effective uses of the computer. In fact, it has been realized that the facilities provided by a time-sharing system's central editing program and its command language are among the most important determinants of the system's convenience, power, and consequent utility. Along the same lines, special-purpose computer-assisted text editing packages have become accepted in industry and government for the preparation or printing of technical manuals, proposals, and other documents in which many updates are necessary and revision time is at a premium."[1]

It, therefore, seems evident that some form of text editing package would be extremely beneficial in the preparation of documentation for computer programs. The ability to produce informative manuals (user guides, operators manuals, maintenance procedures, etc.), descriptions of the methods employed, formal proposals, at minimal cost and effort, with an increase in efficiency, is no longer a nicety, but has now become a necessity.

---

[1] Van Dam, Andries and David E. Rice, "On-Line Text Editing: A Survey". ACM Computing Surveys, Vol. 3, pp. 93-114.

Discussion

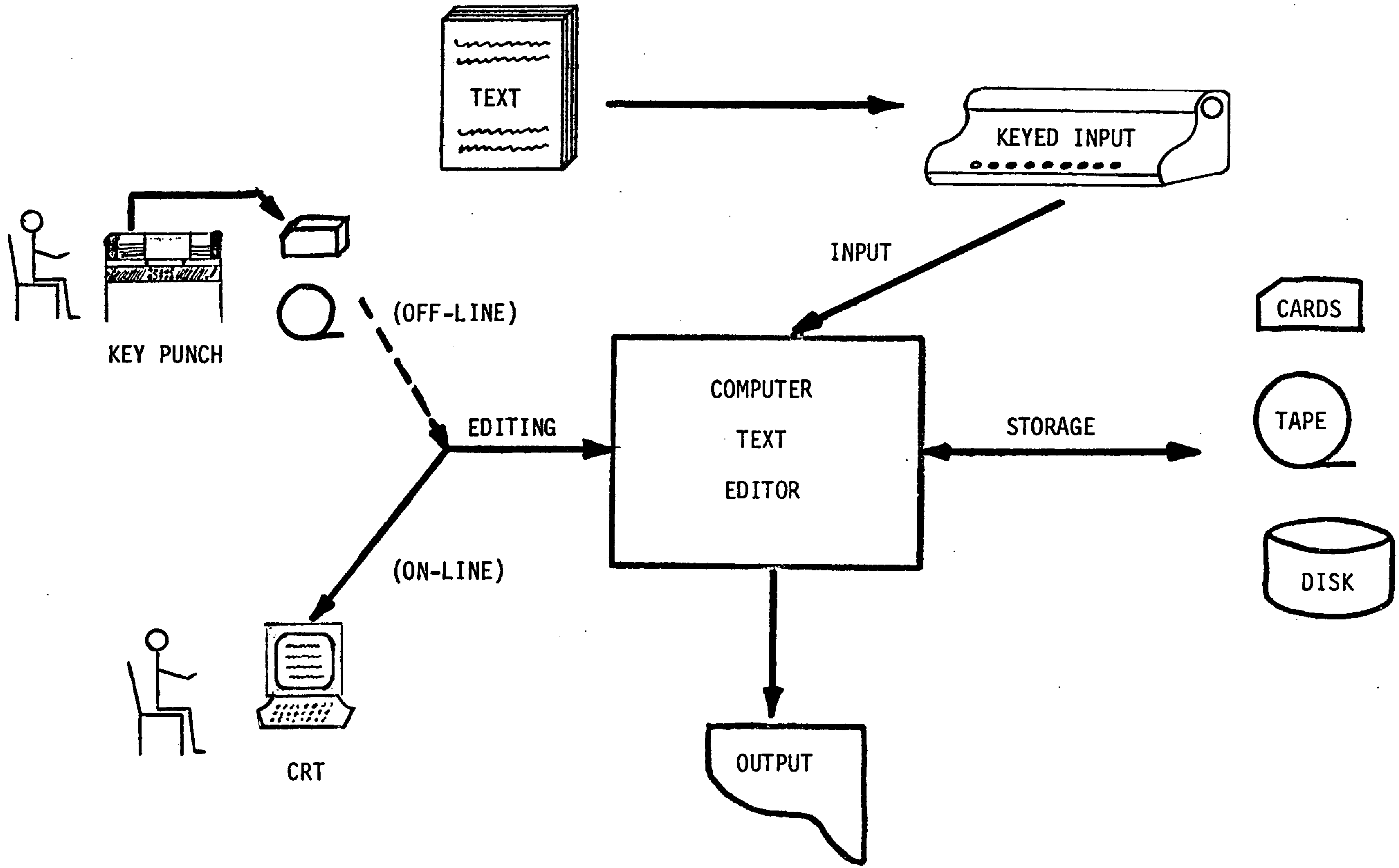
Typically or traditionally, the method of composing some form of text has been merely a typing/proofreading/retyping task - and it is still the same today - but in a slightly modified and extremely more efficient form. Today the editing responsibilities of providing a clean-looking, errorless manuscript have been placed on the computer. This computer text editor still requires much of the same information as the old-fashioned human editor; the text itself, the form or format that it is to be in, plus the knowledge of what is correct and what is not. So in short, the computer text editor is a software package which takes text as input, stores it, modifies it according to the authors' wishes (accomplished by edit codes), and outputs it whenever and however the author specifies.

The sum product of this is the elimination of the communication between author and publisher. The author (programmer or system designer) is now capable of producing and publishing his own works with a tremendous reduction of time and effort. This would enable the author of a computer program to fully document or explain his program with much less effort and frustration than before. Furthermore, if his program were altered in any way, additions or deletions of documentation would only be required in the appropriate places, the additions proof-read, and the entire document reprinted on high-speed line printers or the like. This capability of using high-speed line printers also provides an efficient means of producing many hard-copies of the authors' work for distribution.

Structure of a Text Editor

The typical structure of a text editor is briefly outlined in Figure 1. Several different forms can be used to input the text (keypunched, typewritten code, etc.). This input is generally decoded and fed into the computer text editor. If work is needed on the text, it can now be accomplished or it can be accomplished at a previous or prior time, depending on the type of text editor. Then the desired output is specified and printed or stored for later work or printing.

FIGURE 1



1.) TELETYPEWRITER

2.) LINE PRINTER

Systems Reports

NAME OF SYSTEM	APL TEXT EDITOR
PRICE	NOT APPLICABLE
ON-LINE/OFF-LINE	ON-LINE
MACHINE USED ON	IBM/360, CDC 6600
LANGUAGE WRITTEN IN	NOT AVAILABLE
INPUT FORM(S)	TELETYPED
INPUT DEVICE(S)	IBM 2741 COMMUNICATIONS TERMINAL
OUTPUT FORM(S)	TELETYPEWRITTEN
OUTPUT DEVICE(S)	IBM 2741 COMMUNICATIONS TERMINAL

DISCUSSION:

The APL/TEXT EDITOR is used primarily for form letters or short reports. A limiting factor in the size of the data base is determined by the amount of space allocated to the APL workspace.

All input and output is through the IBM 2741 Communications Terminal. This means that the input rate is determined by the speed of the typist; while the output is at the rate of 140 words per minute.

The system is capable of producing well formatted text (one column only) that can be right justified if specified. Typical usage has shown that less than five pages of text is the maximum amount easily handled. The ability to change typing elements provides for flexibility in fonts and styles of type. This would be valuable for short scientific papers. Text can also be added at any point of the text, thus enabling its usage for form letters.

The edit commands and rules for this system are relatively simple and easy to learn. Experience has shown that only an hour or two is

required to learn how to use the system efficiently. (See chart for list of capabilities.)

COMMENTS:

The best use of this system would be to provide small reports on the status of computer programs or brief overall characteristics (general documentation) for a global view.

It is one of the better small text editing systems.

NAME OF SYSTEM	MAGNETIC TAPE SELECTRIC TYPEWRITER (MT/ST)
PRICE	NOT AVAILABLE
ON-LINE/OFF-LINE	OFF-LINE
MACHINE USED ON	SMALL CONTROL/MEMORY UNIT
LANGUAGE WRITTEN IN	NOT APPLICABLE
INPUT FORM(S)	TELETYPEWRITTEN, (TAPE)
INPUT DEVICE(S)	IBM 2741 Communications Terminal
OUTPUT FORM(S)	TELETYPEWRITTEN
OUTPUT DEVICE(S)	IBM 2741 Communication Terminal

DISCUSSION:

The MT/ST provides neat-looking (but unjustified) form letters and manuscripts. It is primarily used for documents of high-use and low-change rate.

The only editing allowed in MT/ST is to substitute equal length character strings; to effect a substitution of a larger for a smaller string, the entire manuscript with the correction must be copied onto a second tape. The correct place for an edit is located by printing out the contents of the tape until the area of the edit is approached; the operator must then manually stop the processor and retype the line.

Printing is done on the typewriter at the rate of 150 words per minute, and various fonts and type sizes may be used by changing the type ball.

COMMENTS:

The cost and storage of tapes plus the intricacies of revision

(but done by the original typist) are two points which must be considered.

For technical documents which are single column format, use of the MT/ST makes correcting easy; once the document is completed, the tapes can be erased and reused. The MT/ST would be valuable if it is used primarily as a production tool and not as a storage device.

NAME OF SYSTEM	ASTROCOMP
PRICE	NOT AVAILABLE
ON-LINE/OFF-LINE	OFF-LINE
MACHINE USED ON	DEC PDB8L 8K MINICOMPUTER
LANGUAGE WRITTEN IN	NOT AVAILABLE
INPUT FORM(S)	TELETYPEWRITTEN, (TAPE)
INPUT DEVICE(S)	TELETYPEWRITER
OUTPUT FORM(S)	TELETYPEWRITTEN, PHOTO-COMPOSITION
OUTPUT DEVICE(S)	TELETYPEWRITER, PHOTO-COMPOSITER

DISCUSSION:

The ASTROCOMP system is similar to MT/ST in its intended use (small amounts of text, form letters, etc.). This system does have an added advantage of having up to four typewriters connected to a single control unit (DEC PDP-8).

The basic editing command is SUBSTITUTE. As in the APL and MT/ST systems, the "old" (text to be replaced) is typed in to locate it (thus uniquely identifying the text) and the "new" text is then added - replacing the old text.

This system is slightly more powerful than either APL or MT/ST and is a bit more reasonable than MT/ST in its editing features.

It is capable of producing output in photo-composition form, which is a feature neither APL nor MT/ST have.

NAME OF SYSTEM	CALL/360: DATA TEXT
PRICE	NOT AVAILABLE
ON-LINE/OFF-LINE	ON-LINE
MACHINE USED ON	IBM 360
LANGUAGE WRITTEN IN	NOT AVAILABLE
INPUT FORM(S)	TELETYPEWRITTEN
INPUT DEVICE(S)	IBM 2741
OUTPUT FORM(S)	TELETYPEWRITTEN
OUTPUT DEVICE(S)	IBM 2741

DISCUSSION:

DATATEXT is a terminal-oriented, on-line system for data entry, change, and retrieval. It is comprised of an IBM 2741 Communications Terminal in the user's office, phone linking connections, and an IBM 360.

The system is comparable to the TEXT 360 system as far as most formatting capabilities are concerned, but the DATATEXT system is limited to the typewriter terminal for output. (This means 140 words/minute vs. line-printer speed for TEXT 360.)

However, there is an advantage to having typewriter terminals for output; that being the ability to have the document (or sections of) printed out whenever there is a terminal. This saves time because distribution of the document is not necessary.

NAME OF SYSTEM	EDIT
PRICE	NOT AVAILABLE
ON-LINE/OFF-LINE	ON-LINE
MACHINE USED ON	CDC 6000
LANGUAGE WRITTEN IN	NOT AVAILABLE
INPUT FORM(S)	TYPEWRITTEN, CRT KEYED
INPUT DEVICE(S)	TYPEWRITTEN, CRT
OUTPUT FORM(S)	TYPEWRITTEN, CRT DISPLAY
OUTPUT DEVICE(S)	TYPEWRITTEN, CRT

DISCUSSION:

The Text Editor (EDIT) subsystem executer data file manipulations specified by the time-sharing terminal user. These manipulations are performed on a new file or a data file which has been saved in the KRONOS permanent file system.

EDIT allows the user to edit a data file. The data file being edited is known as the search file. During editing, the search-pointer identifies the line of the search file that is currently accessible. The search-pointer can be moved forward and backward during editing to specify a new line. The search-pointer is always positioned at the beginning of a line.

NAME OF SYSTEM	ED PROCESSOR
PRICE	NOT AVAILABLE
ON-LINE/OFF-LINE	ON-LINE
MACHINE USED ON	UNIVAC 1100
LANGUAGE WRITTEN IN	NOT AVAILABLE
INPUT FORM(S)	TELETYPEWRITTEN
INPUT DEVICE(S)	TELETYPEWRITTER
OUTPUT FORM(S)	TELETYPEWRITTEN, PRINTED, CARDS
OUTPUT DEVICE(S)	TTY, PRINTER, CARD-PUNCH

DISCUSSION:

The ED processor proceeds sequentially through the text. It is therefore more efficient to perform editing operation in a more or less sequential manner starting at the beginning of the text.

There are certain processes within the editor which if indiscriminately interrupted can cause the processor to fail. To protect against this, the processor is designed to break at certain points when it is safe to do so.

Comments:

Not much information on this text editor was available at the present time. However, it appears to be below average as far as flexibility and ease of learning is concerned when compared with other text editor systems.

NAME OF SYSTEM	ADMINISTRATIVE TERMINAL SYSTEM (ATS)
PRICE	NOT AVAILABLE
ON-LINE/OFF-LINE	ON-LINE
MACHINE USED ON	IBM/360
LANGUAGE WRITTEN IN	IBM 360 ASSEMBLY
INPUT FORM(S)	TYPED, PUNCHED
INPUT DEVICE(S)	TYPEWRITER KEYBOARD, CARD READER, MAGNETIC TAPE
OUTPUT FORM(S)	TYPED, PUNCHED
OUTPUT DEVICE(S)	IBM 2741, CARD PUNCH, LINE-PRINTER

DISCUSSION:

This system consists of control and functional programs that permit many different text-processing and data-handling activities to be carried on simultaneously through different typewriter terminals attached to an IBM System/360. Written to operate under OS/360, the Administrative Terminal System runs in a multi-programming environment. It will run concurrently with and independently of other tasks in other partitions/ regions.

COMMENTS:

The ATS system is one of the most versatile text processing system that is available. It is powerful and flexible. The output can be in any conventional form and written out at any time.

This system provides the best approach to solving documentation procedures and requirements in regards to cost, ease, and efficiency, as of present.

P. 4.

NAME OF SYSTEM	TEXT/360
PRICE	FREE FROM IBM
ON-LINE/OFF-LINE	OFF-LINE
MACHINE USED ON	IBM/360
LANGUAGE WRITTEN IN	PL/I IBM 360 ASSEMBLY
INPUT FORM(S)	CARDS
INPUT DEVICE(S)	CARD READER
OUTPUT FORM(S)	CARDS, PRINTED
OUTPUT DEVICE(S)	CARD PUNCH, LINE-PRINTER

DISCUSSION:

The TEXT/360 system is strictly an off-line text editor. All text is initially read into the computer via keypunched cards. Edit codes are embedded in this text prior to the input.

Once the cards have been read in, the text is stored on magnetic tape. Any updating or editing after this is also done by cards, using slightly different codes.

This is rather cumbersome and time-consuming for the initial setup, but the power of this system comes from the unlimited amount of text that can be read in, the relative ease of updating, and the elaborate printed output formats for the text.

The only feature <u>not</u> readily available on this system is the ability to change fonts and type style due to the problems involved in changing print chains.

COMMENTS:

This is a well-established and widely used system. Its principle

value comes from the wide range of features available. Also, the only expense to a computer installation would be the running of the software package itself. It requires no special equipment and on-line connection expense. It is also very easy to learn and use.

The system provides excellent text editing capabilities for typical text production needs.

NAME OF SYSTEM	HYPERTEXT EDITING SYSTEM (HES)
PRICE	FREE FROM IBM
ON-LINE/OFF-LINE	ON-LINE
MACHINE USED ON	IBM S/360 WITH IBM 2250 CRT
LANGUAGE WRITTEN IN	IBM 360 ASSEMBLY
INPUT FORM(S)	KEYED INPUT
INPUT DEVICE(S)	IBM 2250 (CRT), LIGHT PEN
OUTPUT FORM(S)	CRT DISPLAY, TYPEWRITTEN
OUTPUT DEVICE(S)	IBM 2741 Communications Terminal, Line Printer

DISCUSSION:

The Hypertext Editing System is a flexible, CRT-based system allowing full editing and formatting capabilities. It is oriented towards "typeset" output (using a computer line printer) as well as flexible input and on-line editing. A lightpen and a set of "function keys", under program control, are used to indicate to the system the nature of the edit to be performed. The portion (s) to text to which the function applies are then indicated by pointing at the text with the lightpen. No command codes for the functions need be remembered, and no extra typing is required to indicate a context string.

Editing commands are INSERT, DELETE, SUBSTITUTE, REARRANGE, and copy.

Many formatting options are available so that text may be formatted both for on-line display and hard-copy printouts. Usually the TEXT/360 program is used for final hard-copy printing.

HES has a unique, however, complicated data structure. A practical

example of a HYPERTEXT system might be an on-line encyclopedia or a set of programming and systems reference manuals, with each cross-reference lightpen sensitive.

COMMENTS:

Even though many nice features are available through this system - easy editing by lightpen and random accessing to any point within the text, providing efficient time-saving updates, it is more costly due to the expensive CRT with lightpen facilities and large amounts of computer time required to operate the system.

NAME OF SYSTEM	FILE RETRIEVAL AND EDITING SYSTEM (FRESS)
PRICE	$35,000 or $500/month
ON-LINE/OFF-LINE	ON-LINE
MACHINE USED ON	IBM/360
LANGUAGE WRITTEN IN	NOT AVAILABLE
INPUT FORM(S)	TELETYPEWRITTEN, KEYED
INPUT DEVICE(S)	IBM 2741 COMMUNICATIONS TERMINAL IBM 2260 CRT
OUTPUT FORM(S)	TYPEWRITTEN, PHOTO-COMPOSITION
OUTPUT DEVICE(S)	IBM 2741 COMMUNICATIONS TERMINAL LINE PRINTER PHOTO-COMPOSER

DISCUSSION:

FRESS is a sophisticated and cost effective text manipulation system, commercially available, and capable of supporting a spectrum of terminals in such a way that all functions are available even on the lowest power terminals.

FRESS is the production version of the predecessor prototype HES.

Beyond normal editing and formatting commands of most text editors, FRESS includes completely arbitrary size string edits, pattern scanning, keyword retrieval, photo-composition output, interfile linking and editing, and protection of files and blocks of text by passwords.

COMMENTS:

Eventhough the price is somewhat steep, the capabilities and usage of this system far exceed any other system reviewed to date.

The commands are simple, but extremely powerful, and easily learned.

FRESS was designed more in the light of being dynamic than other systems, therefore indicating that the system can be "tailor-made" for the customer.

NAME OF SYSTEM	REDIT/RUNOFF
PRICE	NOT AVAILABLE
ON-LINE/OFF-LINE	ON-LINE
MACHINE USED ON	TSS/360
LANGUAGE WRITTEN IN	NOT AVAILABLE
INPUT FORM(S)	TELETYPEWRITTEN
INPUT DEVICE(S)	TELETYPEWRITTEN
OUTPUT FORM(S)	TELETYPEWRITTEN, PRINTED
OUTPUT DEVICE(S)	TELETYPEWRITTEN, LINE-PRINTER

DISCUSSION:

A terminal is connected to TSS/360 by telephone lines of text are typed on the terminal, transmitted to TSS/360 and permanently stored. The text is entered without concern for staying within certain margins or running of the bottom of the page. Special format control words are included in the document text which informs the document printer as to margins, size of paper, spacing, paragraphs, etc.

Delete character and line symbols allow typing errors to be corrected as they are made. Typing errors which have been stored are easily corrected with the text editor. In addition a document can be revised by adding, deliting or re-arranging sections of text.

The document can be printed at the terminal at various stages of revision.

COMMENTS:

As far as time-sharing editing systems are concerned, this is one of the better thus far reviewed. It provides powerful commands and is relatively easy to learn.

NAME OF SYSTEM	TEXT
PRICE	Not Available
ON-LINE/OFF-LINE	ON-LINE
MACHINE USED ON	Sigma 6/7/9
LANGUAGE WRITTEN IN	Not Available
INPUT FORM(S)	Teletypwritten
INPUT DEVICE(S)	Teletypwritten
OUTPUT FORM(S)	Teletypwritten
OUTPUT DEVICE(S)	Teletypwritter, Printer

DISCUSSION:

TEXT is Xerox's text editing system. It is an on-line system that is quite similar to ATS and other line text editors.

The only problem that is immediately evident is the fact that it is designed to operate on the Sigma computers, which may hinder using TEXT as a documentation aid for producing automatic program documentation for an programs on any machine.

Comments:

Disregarding other factors and considering only text editing capabilities, TEXT is a very powerful and capable text editing system. The commands are simple and it is easy to learn how to use. Definitely it should be high on the list of available text editing systems.

NAME OF SYSTEM	NLS; TNLS (TYPEWRITER NLS); DNLS (DISPLAY NLS)
PRICE	NOT CURRENTLY AVAILABLE OUTSIDE THE ARPA NETWORK
ON-LINE/OFF-LINE	ON-LINE
MACHINE USED ON	DEC PDP 10
LANGUAGE WRITTEN IN	UNAVAILABLE
INPUT FORM(S)	TELETYPE, CRT
INPUT DEVICE(S)	ANY TERMINAL ACCEPTABLE TO ARPA NETWORK
OUTPUT FORM(S)	TELETYPE, CRT, PRINTER
OUTPUT DEVICE(S)	ANY DEVICE ACCEPTABLE TO ARPA NETWORK

DISCUSSION:

The NLS text manipulation and editing system were developed by the Augmentation Research Center of Stanford Research Institute. NLS is used as part of ARPA Network Information Center (NIC). NIC performs information retrieval functions.

When working with NLS through a terminal one is at all times constructing, studying, or modifying a file. NLS files have a hierarchical, tree, or outline structure.

NLS has commands which manipulate data on a file level and on the text level. On the file level data is loaded, updated, verified, output and locked or unlocked all through terminal commands. On the text level it is possible to access data in the hierarchy directly by giving a tree address, or relatively according to the current portion being viewed. Commands allow the user to proceed to the next item at the same level,

forward or back, to go up levels or down levels. Text is stored and viewed in upper and lower case founts.

Data is easily updated, move , copied or deleted. It can be viewed on any level with easy transition between levels. Output to any device can be controlled from the terminal.

The NLS system can be used with any terminal accepted by the ARPA Network or with the Special terminal developed by the Augumentation Research Center which uses a 5-key keyboard and a "mouse" to control the cursor in addition to the normal terminal keyboard.

COMMENTS:

This is perhaps the most powerful and versatile text editor reviewed. It's use is at present limited to members of the ARPA Network and personnel at SRI.

Conclusion

Computer aided editing of text has become an established cost-effective use of computers. There exists as wide a range of text editors as there are varieties of text. For small amounts of text, such as memoranda, form letters, brief descriptions, or update reports, systems like the MT/ST, APL text editor, or ASTROCOMP are effective and money-saving. However, for large amounts of text that require more storage, more intricate formats, etc., to produce program descriptions, user manuals, etc., systems like the ATS, TEXT/360, NLS, FRESS would produce the best results.

Any of the systems mentioned previously can be extremely valuable to anyone who uses them, provided that their requirements are in accordance to the system's capabilities and limitations. It is sometimes advisable to employ a combination of these systems to accomplish any desired task.

The principle advantage is not a direct result of these capabilities. It is the indirect effect it has on the documentor. For example, the typical computer programmer enjoys the programming itself, but lacks the desire to sit down and tediously write good documentation for it. The inability to use the computer to help with this task frustrates and annoys him. But with a computer text editor, he is able to type in any information or explanation he likes, changes it at a later date if it is not what he wants, and provides himself and his co-workers with a good, well-formatted and well-documented explanation of what he has done, all with minimal effort.

It can therefore be concluded, that a computer aided text editing system can cut the cost of printed material, stimulate the desire to document, and provide a means to keep the documentation up-to-date.

Foreword to Table I

The following table lists the most of capabilities of text editors. Each text editor feature is briefly described.

In most cases these capabilities are given mnemonic codes which are instructions to the text editor. They are most commonly embedded within the input text itself.

These commands represent the power of the system to update text and to produce formatted text documents.

TABLE I

Capabilities of the Systems

1) TEXT COLUMNS - the ability to have text printed in one or two columns.

2) PAGE/TEXT WIDTH - the ability to specify the number of characters per line; either one or two columns.

3) PAGE/TEXT DEPTH - the ability to specify the number of lines on a page (normal range: approximately 25 to 75 lines).

4) NEW PAGE/NEW COLUMN - for shipping to a new page/column or several pages/columns.

5) RUNNING HEADS AND FEET - allows for running heads (title, subtitle, date, etc.) and running feet, to be printed on each page, only even-numbered, or only odd numbered.

6) RIGHT-HAND PAGES - allows for text to be only printed on one side (right-handed).

7) "AS-IS" TEXT - allows for text to be printed in any format that is inputed (blanks are not squeezed out, or added to aid in right justification).

8) CENTERED TEXT - provides for specified text to be centered, equally between right and left hand margins.

9) LINE-SPACING - specifies single or double-spaced printing.

10) BLANK LINES - provides for skipping lines.

11) PARAGRAPHS - specifies that a line should be skipped and indented a specified or default number of spaces.

12) INDENTIONS - allows for a specified indention for any line of text.

13) HANGING INDENTIONS - allows for the text to be indented for as many lines as specified.

14) COLUMN JUSTIFICATION - causes all the text columns on a page to be equal in length by spacing the columns out to the page depth.

15) LINE JUSTIFICATION - causes all text lines to be spaced out to the line length by the insertion of blank characters between words.

16) TABULAR TEXT - specifies tab settings and indicates the text to be printed at the settings.

17) HYPHENATION - allows the program to hyphenate the last word on a line if needed for justification purposes.

18) FOOTNOTES - allows specified text that is to be printed as a footnote.

19) HEADINGS - provides for line(s) of text that are set apart from the text body and provides a description of the section that follows it.

20) KEEPS - specifies a portion of text, such as a table, heading, or the like, so that it is not split between two columns or pages.

21) TABLES - allows the preparation of tables and charts within the text.

22) HORIZONTAL LINES - provides the ability to print horizontal lines.

23) VERTICAL LINES - provides the ability to print vertical lines.

24) CAPITALIZATION - allows capital letters.

25) UNDERSCORING - allows the underscoring of any text.

26) USER-DEFINED CODES - permits the user to define his own code which can be comprise of combinations of regular edit codes.(This capability is generally used for repetitive edit functions)

27) SUPPLEMENTAL LISTINGS - provides specified text to be printed at specific times. (generally used to create Table of Contents, list of figures, etc.)

28) MULTIPLE PRINTING OF TEXT - allows specified pages or documents to be printed more than once at any given time.

29) REVISION BARS - allows updated text to be indicated by a vertical bar in the margin, signifying that portion of text has been revised.

30) RENUMBERING - permits the automatic renumbering of pages in a document or report after updating (addition or deletion of text).

31) REPAGING - permits the automatic repaging of text.

32) VARIABLE STYLES - allows for the use of special characters within a text. Normally this is limited to the type of print chain available or the kinds of typing elements available.

Foreword to Table II

This table lists what are commonly referred to as the edit commands. These commands differ from the format commands in the fact that they are used to manipulate portions of text to make changes, deletions or insertions within a document instead of formatting the text itself.

These commands generally vary the most in format from one system to the next, but all systems can accomplish these basic functions.

TABLE II

Editing Features

1) ADDITION/DELETION OF CHARACTERS
2) ADDITION/DELETION OF LINES
3) ADDITION/DELETION OF PARAGRAPHS
4) ADDITION/DELETION OF PAGES
5) TRANSFERING CHARACTERS
6) TRANSFERING LINES
7) TRANSFERING PARAGRAPHS
8) TRANSFERING PAGES

NAME OF SYSTEM	APL	MTST	ASTROCOMP
ON-LINE/OFF-LINE	On-Line	Off-Line	Off-Line
INPUT FORM(S)	Typed	Typed, Tape	Typed, Tape
OUTPUT FORM(S)	Typed	Typed	Typed Photo-composition
EDITING FEATURES			
ADD/DELETE CHARACTERS	Yes	Yes	No
ADD/DELETE WORDS	Yes	Yes	No
ADD/DELETE LINES	Yes	Yes	Yes
ADD/DELETE PARAGRAPHS	Yes	Yes	Yes
ADD/DELETE PAGES	Yes	Yes	Yes
TRANSFER WORDS	Yes	No	No
TRANSFER PARAGRAPHS	No	No	Yes
TRANSFER PAGES	No	No	Yes
FORMAT FEATURES			
TEXT COLUMNS	No	No	Yes
PAGE/TEXT WIDTH	Yes	Yes	Yes
PAGE/TEXT DEPTH	Yes	Yes	Yes
NEW PAGE/NEW COLUMN	Yes/No	Yes/No	Yes/No
RUNNING HEADS AND FEET	No	No	No
RIGHT-HAND PAGES	No	No	No
"AS-IS TEXT"	No	Yes	No
CENTERED TEXT	Yes	Yes	Yes
LINE-SPACING	Yes	Yes	Yes
BLANK LINES	Yes	Yes	Yes
PARAGRAPHS	Yes	Yes	Yes
INDENTIONS	Yes	Yes	Yes

NAME OF SYSTEM	APL	MTST	ASTROCOMP
FORMAT FEATURES (cont.)			
HANGING INDENTIONS	Yes	Yes	Yes
COLUMN JUSTIFICATION	No	No	No
LINE JUSTIFICATION	Yes	No	Yes
TABULAR TEXT	No	No	Yes
HYPHENATION	No	No	Yes
FOOTNOTES	Yes	Yes	Yes
HEADINGS	Yes	Yes	No
KEEPS	No	No	Yes
TABLES	No	No	Yes
HORIZONTAL LINES	No	No	Yes
VERTICAL LINES	No	No	Yes
CAPITALIZATION	Yes	Yes	Yes
UNDERSCORING	Yes	Yes	Yes
USER-DEFINED CODES	No	No	No
SUPPLEMENTAL LISTINGS	No	No	No
MULTIPLE PRINTINGS	Yes	Yes	Yes
REVISION BARS	No	No	No
RENUMBERING	Yes	Yes	Yes
REPAGING	No	No	No
VARIABLE STYLES	Yes	Yes	Yes

NAME OF SYSTEM	DATA TEXT	EDIT	ED PROCESSOR
ON-LINE/OFF-LINE	On-Line	On-Line	On-Line
INPUT FORM(S)	TTY	TTY	CRT
OUTPUT FORM(S)	TYPED, PRINTED	TYPED	CRT DISPLAY, PRINTED
EDITING FEATURES	Yes	Yes	Yes
ADD/DELETE CHARACTERS	Yes	Yes	Yes
ADD/DELETE WORDS	Yes	Yes	Yes
ADD/DELETE LINES	Yes	Yes	Yes
ADD/DELETE PARAGRAPHS	Yes	Yes	Yes
ADD/DELETE PAGES	Yes	Yes	Yes
TRANSFER WORDS	Yes	Yes	Yes
TRANSFER PARAGRAPHS	Yes	Yes	Yes
TRANSFER PAGES			
FORMAT FEATURES			
TEXT COLUMNS	Yes	No	No
PAGE/TEXT WIDTH	Yes	Yes	No
PAGE/TEXT DEPTH	Yes	Yes	No
NEW PAGE/NEW COLUMN	Yes	Yes/No	Yes/No
RUNNING HEADS AND FEET	Yes	No	No
RIGHT-HAND PAGES	Not Available	No	No
"AS-IS TEXT"	Yes	No	No
CENTERED TEXT	Yes	Yes	Yes
LINE-SPACING	Yes	Yes	Yes
BLANK LINES	Yes	Yes	Yes
PARAGRAPHS	Yes	Yes	Yes
INDENTIONS	Yes	Yes	Yes

NAME OF SYSTEM	DATATEXT	EDIT	ED PROCESSOR
FORMAT FEATURES (cont.)			
HANGING INDENTIONS	Yes	Yes	Yes
COLUMN JUSTIFICATION	Yes	No	No
LINE JUSTIFICATION	Yes	No	No
TABULAR TEXT	Yes	No	No
HYPHENATION	Not Available	No	No
FOOTNOTES	Yes	No	No
HEADINGS	Yes	Yes	Yes
KEEPS	Not Available	No	No
TABLES	Yes	No	No
HORIZONTAL LINES	Not Available	No	No
VERTICAL LINES	Not Available	No	No
CAPITALIZATION	Yes	Yes	Yes
UNDERSCORING	Yes	Yes	Yes
USER-DEFINED CODES	Not Available	Not Available	No
SUPPLEMENTAL LISTINGS	Not Available	Not Available	No
MULTIPLE PRINTINGS	Yes	Yes	Not Available
REVISION BARS	Not Available	No	No
RENUMBERING	Yes	No	No
REPAGING	Yes	No	No
VARIABLE STYLES	Yes	Yes	No

NAME OF SYSTEM	ATS	TEXT/360	HES
ON-LINE/OFF-LINE	ON-LINE	OFF-LINE	ON-LINE
INPUT FORM(S)	TTY	CARDS	KEYED INPUT
OUTPUT FORM(S)	TYPED, PRINTED, CARDS, TAPE	PRINTED	CRT DISPLAY, TYPED
EDITING FEATURES	Yes	Yes	Yes
ADD/DELETE CHARACTERS	Yes	Yes	Yes
ADD/DELETE WORDS	Yes	Yes	Yes
ADD/DELETE LINES	Yes	Yes	Yes
ADD/DELETE PARAGRAPHS	Yes	Yes	Yes
ADD/DELETE PAGES	Yes	Yes	Yes
TRANSFER WORDS	Yes	Yes	Yes
TRANSFER PARAGRAPHS	Yes	Yes	Yes
TRANSFER PAGES	Yes	Yes	Yes
FORMAT FEATURES			
TEXT COLUMNS	Yes	Yes	No
PAGE/TEXT WIDTH	Yes	Yes	Yes
PAGE/TEXT DEPTH	Yes	Yes	Yes
NEW PAGE/NEW COLUMN	Yes	Yes	Yes/No
RUNNING HEADS AND FEET	Yes	Yes	Not Available
RIGHT-HAND PAGES	Yes	Yes	Not Available
"AS-IS TEXT"	Yes	Yes	Not Available
CENTERED TEXT	Yes	Yes	Yes
LINE-SPACING	Yes	Yes	Yes
BLANK LINES	Yes	Yes	Yes
PARAGRAPHS	Yes	Yes	Yes
INDENTIONS	Yes	Yes	Yes

NAME OF SYSTEM	ATS	TEXT/360	HES
FORMAT FEATURES (cont.)	Yes	Yes	No
HANGING INDENTIONS	Yes	Yes	No
COLUMN JUSTIFICATION	Yes	Yes	Yes
LINE JUSTIFICATION	Yes	Yes	Not Available
TABULAR TEXT	Yes	Yes	Yes
HYPHENATION	Yes	Yes	No
FOOTNOTES	Yes	Yes	Yes
HEADINGS	Yes	Yes	Yes
KEEPS	Not Available	Yes	Not Available
TABLES	Yes	Yes	Not Available
HORIZONTAL LINES	Not Available	Yes	Not Available
VERTICAL LINES	Not Available	Yes	Not Available
CAPITALIZATION	Yes	Yes	Yes
UNDERSCORING	Yes	Yes	Yes
USER-DEFINED CODES	Not Available	Yes	Not Available
SUPPLEMENTAL LISTINGS	Not Available	Yes	Not Available
MULTIPLE PRINTINGS	Yes	Yes	Yes
REVISION BARS	Not Available	Yes	Not Available
RENUMBERING	Yes	Yes	Yes
REPAGING	Yes	Yes	Yes
VARIABLE STYLES	Yes	Limited to Type-Chains	No

NAME OF SYSTEM	FRESS	REDIT/RUNOFF	TEXT
ON-LINE/OFF-LINE	ON-LINE	ON-LINE	ON-LINE
INPUT FORM(S)	KEYED INPUT	TTY	TTY
OUTPUT FORM(S)	CRT DISPLAY PHOTO-COMPOSITION	PRINTED TELETYPEWRITTEN	PRINTED TELETYPWRITTEN
EDITING FEATURES			
ADD/DELETE CHARACTERS	Yes	Yes	Yes
ADD/DELETE WORDS	Yes	Yes	Yes
ADD/DELETE LINES	Yes	Yes	Yes
ADD/DELETE PARAGRAPHS	Yes	Yes	Yes
ADD/DELETE PAGES	Yes	Yes	Yes
TRANSFER WORDS	Yes	Yes	Yes
TRANSFER PARAGRAPHS	Yes	Yes	Yes
TRANSFER PAGES	Yes	Yes	Yes
FORMAT FEATURES			
TEXT COLUMNS	Yes	Yes	No
PAGE/TEXT WIDTH	Yes	Yes	Yes
PAGE/TEXT DEPTH	Yes	Yes	Yes
NEW PAGE/NEW COLUMN	Yes	Yes	Yes/No
RUNNING HEADS AND FEET	Yes	Yes	Yes
RIGHT-HAND PAGES	Yes	Yes	Yes
"AS-IS TEXT"	Yes	Not Available	Yes
CENTERED TEXT	Yes	Yes	Yes
LINE-SPACING	Yes	Yes	Yes
BLANK LINES	Yes	Yes	Yes
PARAGRAPHS	Yes	Yes	Yes
INDENTIONS	Yes	Yes	Yes

NAME OF SYSTEM	FRESS	REDIT/RUNOFF	TEXT
FORMAT FEATURES (cont.)	Yes	Yes	Yes
HANGING INDENTIONS	Yes	Yes	No
COLUMN JUSTIFICATION	Yes	Yes	Yes
LINE JUSTIFICATION	Yes	Yes	Yes
TABULAR TEXT	Yes	Yes	Yes
HYPHENATION	Yes	Yes	Yes
FOOTNOTES	Yes	Yes	Yes
HEADINGS	Yes	Yes	Yes
KEEPS	Yes	Not Available	Yes
TABLES	Yes	Not Available	Yes
HORIZONTAL LINES	Yes	Not Available	Yes
VERTICAL LINES	Yes	Not Available	Yes
CAPITALIZATION	Yes	Yes	Yes
UNDERSCORING	Yes	Yes	Yes
USER-DEFINED CODES	Yes	Not Available	No
SUPPLEMENTAL LISTINGS	Yes	Yes	Yes
MULTIPLE PRINTINGS	Yes	Yes	Yes
REVISION BARS	Yes	Yes	No
RENUMBERING	Yes	Yes	Yes
REPAGING	Yes	Yes	Yes
VARIABLE STYLES	Yes	Yes	Yes

NAME OF SYSTEM	NLS, TNLS, DNLS		
ON-LINE/OFF-LINE	ON-LINE		
INPUT FORM(S)	TTY, CRT		
OUTPUT FORM(S)	Typed, CRT, printed		
EDITING FEATURES	YES		
ADD/DELETE CHARACTERS	YES		
ADD/DELETE WORDS	YES		
ADD/DELETE LINES	YES		
ADD/DELETE PARAGRAPHS	YES		
ADD/DELETE PAGES	YES		
TRANSFER WORDS	YES		
TRANSFER PARAGRAPHS	YES		
TRANSFER PAGES	YES		
FORMAT FEATURES			
TEXT COLUMNS	NO		
PAGE/TEXT WIDTH	YES		
PAGE/TEXT DEPTH	YES		
NEW PAGE/NEW COLUMN	YES		
RUNNING HEADS AND FEET	YES		
RIGHT-HAND PAGES	NO		
"AS-IS TEXT"	YES		
CENTERED TEXT	YES		
LINE-SPACING	YES		
BLANK LINES	YES		
PARAGRAPHS	YES		
INDENTIONS	YES		

NAME OF SYSTEM	NLS, TNLS, DNLS		
FORMAT FEATURES (cont.)			
HANGING INDENTIONS	YES		
COLUMN JUSTIFICATION	NO		
LINE JUSTIFICATION	NO		
TABULAR TEXT	YES		
HYPHENATION	NO		
FOOTNOTES	NO		
HEADINGS	YES		
KEEPS	YES		
TABLES	NO		
HORIZONTAL LINES	YES		
VERTICAL LINES	NO		
CAPITALIZATION	YES		
UNDERSCORING	YES		
USER-DEFINED CODES	NO		
SUPPLEMENTAL LISTINGS	YES		
MULTIPLE PRINTINGS	YES		
REVISION BARS	NO		
RENUMBERING	YES		
REPAGING	YES		
VARIABLE STYLES	YES		

F-1

APPENDIX F

PROGRAM EDITORS

AUTOMATIC SYSTEM FOR COMPUTER PROGRAM DOCUMENTATION

F- 1.a

Working Paper No. 3

August 25, 1972

Program Editors

By

Ralph F. Planthold

Texas A&M University

Texas Engineering Experiment Station

F-1-b

ABSTRACT

This report defines a program editor, differentiates between a program editor and free-form text editor, and delineates the two major categories of program editors: on-line and off-line. It shows that the characteristics of on-line program editors may differ because of the particular type of terminal involved.

The report lists the information sources tapped and responses received. Next is a comparison of six on-line editors and a comparison of eleven off-line editors, both in tabular form.

Finally, there is a list of criteria for an ideal program editor and a nomination of the on-line editors and the off-line editors which, in the author's opinion, come closest to satisfying the listed criteria.

F-20

Table of Contents

F-3

INTRODUCTION

A program editor is basically a software package which allows one to modify and document one or more source program modules without the danger of hand-carrying (and possibly dropping or losing) sizable decks of cards and without the attendant need for reading in and compiling the entire source module(s) every time a change is incorporated. As such, a program editor is a rather specialized version of a text editor (a software package for the manipulation of free-form textual material).

A program editor differs from a text editor in the way its material is stored and displayed. Storage in card image or printer line image format is generally quite adequate for performing the "in place" modification - substitution of one opcode, operand, address, label, etc., for another - of computer source programs. Free-form text editors, because of their more powerful capabilities of arbitrary placement and replacement of arbitrarily sized character strings, require a flexible storage stucture, one capable of a noticeable amount of dynamic expansion and contraction. For this the unit of storage must be at least a "super-line" of several hundred characters, and may even be a paragraph or a whole page.

For the output of a program editor, only an upper case character set is needed; and the text is printed as it was stored, line by line. A text editor is limited if it does not have at least upper and lower case in its character set, because the fairly complex typesetting codes it employs would lose a great deal of their impact in such an environment.

Program editors fall into one of two categories: on-line or off-line, depending on their mode of operation. Off-line editors are automatically self-documenting, providing a hard-copy listing of all changes made during the current run and, optionally, a wide variety of reports covering such areas as a program evolution history and a current file status summary. On-line editors provide no such automated documentation, relying on the individual programmer or another documentation system to perform this function.

Off-line editors permit the creation of entire source modules only, whereas on-line editors allow the user to create new routines as he thinks them up, one statement at a time. Since off-line editors are governed by control cards supplied by the user, an error-handling technique is provided. In most cases, if an error is encountered on a control card, all modifications of the source module to which the control card applies are ignored. One noteworthy exception to this is "Simple", an off-line program editor from Computer Services which terminates the program upon finding any errors in its control cards. On-line editors can immediately alert the user to any errors in control information. Upon correction, the user may then proceed with the editing of the program.

Various on-line editors exhibit differences due to the type of terminal they employ. Those communicating with the user via a teletype/typewriter (TTY) terminal show marked similarities to off-line editors in the method of specifying operations and in the manner in which the text is presented. For this reason, any reference to on-line editors

in the comparison of these methods and manners in the two following paragraphs will apply only to those using cathode-ray tube (CRT) units as terminals.

For an off-line editor, specifying what is to be done and where it is to be done requires a command name or abbreviation together with a location denoted by a line number and/or context string. An on-line (CRT) editor typically requires "pointing" to the target text area via a light-pen, a keyboard-controlled cursor, etc., then modifying the text by (perhaps depressing a function key and) typing the new text right over the old.

With an off-line editor, the user is forced to work from some printed copy, making changes thereon and transcribing those changes, resulting in duplication of effort. Because an on-line (CRT) editor simultaneously displays many lines of text with virtually no time lag, the user can think out and implement desired changes in one operation instead of two.

BACKGROUND

Over the past three months, the following sources have been searched for information on the subject of program editors: Auerbach Computer Technology Software Reports, Journal of the Association for Computing Machinery (ACM), ACM Computing Surveys,ACM Computing Review, Communications of the ACM, U.S. Government Research and Development Reports, Proceedings of the AFIPS Fall and Spring Joint Computer Conferences, the Computer Journal,, the IBM S/360 OS Utilities manual, the Univac 1100 Series Operating System Programmer Reference, and the Control Data 6000 Series Computer Systems KRONOS Text Editor Reference Manual.

The Auerbach Reports carried information about a large number of data manipulation packages, four of which were found to be worthwhile as (off-line) program editors. Two of the four originating firms have sent further literature in response to our request for same; one firm remains untraceable. The ACM Computing Surveys carried information on seven (on-line) program editors. Two operated on certain source languages only (one on JOVIAL, the other on PL/I and Gedanken) and have therefore been omitted*; requests for additional information on the rest remain as yet unanswered.

In addition, letters have been sent to thirteen manufacturers of source program maintenance packages listed in the 1971 edition of the Datamation Industry Directory. Four positive replies have been received as of this date; one other firm has gone out of business.

STATE OF THE ART

Summarized in the following tables are the characteristics of six on-line and eleven off-line program editors. The on-line editors treated are: Conversational Context-Directed Editor (CMS), developed at the IBM Cambridge Scientific Center; WYLBUR, developed at the Stanford Computation Center; Quick Editor (QED), developed by the University of California at Berkeley and revised extensively for commercial use by Com-Share, Inc. - the only editor of either category to maintain additional files of editing changes for testing slightly different versions of a program without modifying or duplicating the original program; Text Editor and Corrector (TECO), developed by the Massachusetts Institute of Technology and Project

---

* All program editors treated in this report will operate on any computer source program, written in any source language.

MAC; TVEDIT, developed by the Stanford Computation Center; and EDIT, developed by Control Data Corporation for its KRONOS Time-sharing System.

The off-line editors treated are: LIBRARIAN, by Applied Data Research, Inc.; SIMPLE, by Computer Services; Card Library On Tape (CLOT), by International Telecontrol Corporation; PROGRAM/MANAGE, by Management Systems Corporation; Card File Maintenance System (CFMS), by Sigma Software Company; PANVALET, by Pansophic Systems, Incorporated; PLUS D/A, by the Cullinane Corporation; the Source Program Library System (SPLIS-II), by the Webster Computer Corporation; IEBUPDAT and IEBUPDTE, two IBM System/360 Operating System Utilities programs; and the ED Processor, a text editor which is part of the Univac EXEC-8 Monitor.

Within the tables, an entry of "Not Given" means: "Has or could have this attribute, but no conclusive information given." An entry of "Not Applicable" means: "Could in no way have this attribute." For insertion of a character string or replacement of a shorter one by a longer one, if the record length is exceeded, an entry of "Truncation" means: "Any excess is ignored and lost," while an entry of "Overflow" means: "Any excess causes no insertion/replacement to be made, and a warning to be issued."

ON-LINE PROGRAM EDITORS

		CMS	WYLBUR
Computer Configuration		IBM S/360/67 under CP/CMS Operating System	IBM S/360/67 under OS/MFT, 1 disk drive
Man-Machine Interface		IBM 2741 TTY	IBM 2741 TTY
Storage Requirements (bytes)		Not Given	Not Given
Cost		Not Given	Not Given
Language Written In		Not Given	IBM S/360 BAL
Target Record Type		Fixed Length Line	Variable Length Line
Target Record Size (characters)		80	1-133(Default is 72)
Target Record Processing Fashion		Pages Sequential, Random within Page	Random
Target Text Pinpoint Location Requirements		Relative Line No.	Absolute Line No.
Fixed Position (Label) Scan Capability		Yes	Yes
Arbitrary Position (Context String) Scan Capability		Yes	Yes
Record-Oriented Commands (handle at a time)	Single Record	Insert, Delete, Replace	Insert, Delete, Replace
	Multiple Records	Insert, Delete	Insert, Delete, Replace
String-Oriented Commands (operate within)	Single Record	Insert, Delete, Replace (Truncation)	Insert, Delete, Replace (Truncation)
	Limited Record Range	Replace (Truncation)	Replace (Truncation)
	Entire Record Range (Universal)	Replace (Truncation)	Replace (Truncation)
Temporary Edit Capability for Testing		No	No
Machine-Readable Output	Updated Master File	Yes	Yes
	Job Stream JCL Retrievable from System or Library	Not Applicable	No
	Job Stream Execution without Operator Intervention	Not Applicable	Yes
	Selected Modules in Punched Deck Form	No	Yes
Hard-Copy Documentation Output	Current Run Permanent Edit Listing	Not Applicable	No
	Current Run Temporary Edit Listing	Not Applicable	Not Applicable
	Updated Module Listing	Not Applicable	Yes
	Program Evolution History	Not Applicable	No
	Current File Status Summary	Not Applicable	Yes
Security Controls		None	None
Additional Capabilities		Tabbing Capability	Tabbing Capability

ON-LINE PROGRAM EDITORS

QED	TECO	TVEDIT	EDIT
Not Given, 1 disk drive	DEC PDP-10, 1 disk drive	IBM S/360/67, 1 disk drive	CDC 6000 series under KRONOS, 1 disk drive
TTY	TTY	CRT	TTY
Not Given	4K	Not Given	Not Applicable
Not Given	Not Given	Not Given	Not Applicable
Not Given	Not Given	Not Given	Not Given
Variable Length Super-Line	Variable Length Page	Variable Length Page	Variable Length Line
1-500	Unlimited	1-128 per line	1-150 (Default is 72)
Random	Pages Sequential, Random within Page	Random	Random
Relative Super-Line No.	Relative Character or Line Displacement	Absolute Page No., followed by Visual Scan	Relative Line No.
Yes	No	No	No
Yes	Yes	No	Yes
Insert, Delete, Replace	Not Applicable	Delete	Insert,Delete,Replace
Insert, Delete, Replace	Not Applicable	Not Applicable	Insert,Delete,Replace
Replace (Truncation)	Insert, Delete	Insert, Delete, Replace (1 character at a time)	Insert,Delete, Replace(Truncation)
Replace (Truncation)	Insert, Delete	Not Applicable	Insert,Delete, Replace(Truncation)
Replace (Truncation)	Insert, Delete	Not Applicable	Not Applicable
Yes	No	No	Yes
Yes	Yes	Yes	Yes
Not Applicable	Not Applicable	Not Applicable	Not Applicable
Not Applicable	Yes	Not Applicable	Not Applicable
No	No	No	No
Not Applicable	Not Applicable	Not Applicable	Not Applicable
Not Applicable	Not Applicable	Not Applicable	Not Applicable
Not Applicable	Not Applicable	Not Applicable	Yes
Not Applicable	Not Applicable	Not Applicable	Not Applicable
Not Applicable	Not Applicable	Not Applicable	Not Applicable
None	None	None	None
None	Arithmetic, Tabbing Capability	Command Repetition Factor	Module Manipulation; Elimination of Excess Blanks

OFF-LINE PROGRAM EDITORS

		LIBRARIAN	SIMPLE
Computer Configuration		IBM S/360 under OS or DOS, 1 disk drive	IBM S/360/30, 370 under OS or DOS, 1 disk drive
Man-Machine Interface		Card Reader/Punch, Printer	Card Reader/Punch, Printer
Storage Requirements (bytes)		59K	65K
Cost		$3600 for 1st 3 years, then $500 per year	$2800 for 1st 3 years, then $100 per year
Language Written In		IBM S/360 BAL	IBM S/360 BAL
Target Record Type		Fixed Length Card Image	Fixed Length Card Image
Target Record Size (characters)		80	80
Target Record Processing Fashion		Sequential	Not Given
Target Text Pinpoint Location Requirements		Absolute Card Sequence No.	Not Given
Fixed Position (Label) Scan Capability		No	No
Arbitrary Position (Context String) Scan Capability		Yes	Yes
Record-Oriented Commands (handle at a time)	Single Record	Insert, Delete, Replace	Insert, Delete, Replace
	Multiple Records	Insert, Delete, Replace	Not Given
String-Oriented Commands (operate within)	Single Record	Insert, Replace (Overflow)	Replace (Truncation)
	Limited Record Range	Insert, Replace (Overflow)	Not Given
	Entire Record Range (Universal)	Insert, Replace (Overflow)	Replace (Truncation)
Temporary Edit Capability for Testing		Yes	No
Machine-Readable Output	Updated Master File	Yes	Yes
	Job Stream JCL Retrievable from System or Library	Yes	Not Given
	Job Stream Execution without Operator Intervention	Yes (MFT/MVT only)	No
	Selected Modules in Punched Deck Form	Yes	No
Hard-Copy Documentation Output	Current Run Permanent Edit Listing	Yes	Yes
	Current Run Temporary Edit Listing	No	Not Applicable
	Updated Module Listing	Yes	Yes
	Program Evolution History	Yes	Yes
	Current File Status Summary	Yes	Yes
Security Controls		Unchanging, bypassable, pgm-generated password	Password-based scrambled characterset
Additional Capabilities		COBOL (any level) Syntax checker (12K bytes); Resequencing; Module Manipulation	COBOL Macro and Abbreviation Expansion; Module Manipulation; Resequencing

OFF-LINE PROGRAM EDITORS

CLOT	PROGRAM/MANAGE	CFMS	PANVALET
IBM S/360/25 under OS or DOS, 2 tape drives	IBM S/360/30 under OS or DOS, 2 tape drives	IBM S/360, 370 under DOS, 3 tape drives	IBM S/360/22, 370/135 OS or DOS, 1 disk driv
Card Reader, Printer	Card Reader/Punch, Printer	Card Reader/Punch, Printer	Card Reader/Punch, Printer
Not Given	32K	32K	19K
$950 plus optional $10 per year maintenance	$2485 each authorization	$900 plus optional $428 installation charge	$3780 plus optional $600 per year maint. after 3
COBOL	COBOL	COBOL	Not Given
Fixed Length Card Imag	Fixed Length Card Image	Fixed Length Card Image	Fixed Length Card Image
80	80	80	80
Random	Not Given	Not Given	Sequential
Absolute Card Sequence No.	Absolute Card Sequence No.	Not Given	Absolute Card Sequence No.
No	No	No	Yes
No	No	No	Yes
Insert, Delete	Insert, Delete, Replace	Insert, Delete	Insert, Delete, Replace
Insert, Delete	Insert, Delete, Replace	Insert, Delete	Insert, Delete, Replace
Not Applicable	Not Applicable	Not Applicable	Scan and Print only
Not Applicable	Not Applicable	Not Applicable	Scan and Print only
Not Applicable	Not Applicable	Not Applicable	Scan and Print only
No	No	No	Yes
Yes	Yes	Yes	Yes
No	No	No	Yes
No	No	No	Yes
No	Yes	Yes	Yes
Yes	Yes	Yes	Yes
Not Applicable	Not Applicable	Not Applicable	Yes
Yes	Yes	Yes	Yes
No	Yes	Yes	Yes
No	Yes	Yes	Yes
None	None	None	Up to 3 levels of security control codes
Resequencing	Resequencing	None	Module Manipulation; Formatting; Resequencing; Cross Referencing

OFF-LINE PROGRAM EDITORS

PLUS D/A	SPLIS-II	IEBUPDAT and IEBUPDTE	ED
IBM S/360, 370 under OS or DOS, 1 disk drive	IBM S/360/30, 370 under DOS, 1 disk drive	IBM S/360, 370 under OS, 1 disk drive	UNIVAC 1100 series under EXEC-8, 1 disk drive
Card Reader/Punch, Printer	Card Reader/Punch, Printer	Card Reader/Punch, Printer	Card Reader/Punch, Printer
50K	44K	Not Applicable	Not Applicable
$3500 for 1st 3 years, then $525 per year	$2500 1 installn, $2000 2nd, $1500 each additnl	Not Applicable	Not Applicable
IBM S/360 BAL	COBOL	Not Given	Not Given
Fixed Length Card Image	Fixed Length Card Image	Fixed Length Card Image	Fixed Length Line
80	80	80	Not Given (Default 80)
Sequential	Random	Random	Random
Absolute Card Sequence No.	Not Given	Absolute Card Sequence No.	Relative Card Sequence No.
No	No	No	Yes
No	No	No	Yes
Insert, Delete, Replace	Insert, Delete	Insert, Delete, Replace	Insert,Delete,Replace
Insert, Delete, Replace	Insert, Delete	Insert, Delete, Replace	Insert,Delete
Not Applicable	Not Applicable	Not Applicable	Replace
Not Applicable	Not Applicable	Not Applicable	Replace
Not Applicable	Not Applicable	Not Applicable	Replace
Yes	Yes	No	Yes
Yes	Yes	Yes	Yes
Yes	Not Given	Not Applicable	Not Applicable
No	No	Not Applicable	Not Applicable
Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes
No	Yes	Not Applicable	Yes
Yes	Yes	Yes	Yes
Yes	No	No	No
Yes	Yes	No	No
Installation-changeable scrambled character set	Password	None	None
Resequencing	Module Manipulation; Resequencing	Resequencing; Change Data Set Organization (IEBUPDTE only)	Tabbing Capability; Module Manipulation

CONCLUSIONS: THE "IDEAL" PROGRAM EDITOR

While user requirements differ widely, and no one program editor could ever satisfy all potential users, certain desirable characteristics do stand out above the rest. The "ideal" program editor should be capable of:

1. Easy mastery by programmer personnel (a minimal amount of memorization and time required for entry of edit command codes and operands)
2. High-speed response
3. Processing records out of sequence
4. Universal replacement of character strings
5. Inserting a character string or replacing a shorter with a longer without fear of truncation
6. Maintaining files for edit operations, applicable at will to a master file
7. Creating a job stream for compilation/execution of changed modules, with JCL capable of being accepted from the user and/or operating system and/or source program library
8. Producing hard-copy documentation of all permanent and temporary changes made during the current run and a history of all changes ever made to a given module
9. Displaying many lines of text simultaneously, with the ability to display any portion of the entire text very rapidly (requires an on-line editor with a CRT terminal)

Of the program editors treated in this report, the two most nearly satisfying these ideal requirements for both categories are:

1. On-Line - TVEDIT and WYLBUR
2. Off-Line - LIBRARIAN and PANVALET

BIBLIOGRAPHY

Applied Data Research, Inc. Librarian. Princeton: Applied Data Research, Inc., 1972.

Auerbach Associates, Inc. "Data Manipulation", Auerbach Computer Technology Reports: Software Reports, LI (1972), 6007.85-6007.87, 6007.91-6007.93, 6007.107-6007.108, 6008.129-6008.131.

Control Data Corporation. KRONOS Text Editor (EDIT) Reference Manual. Minneapolis: Control Data Corporation, 1971.

Cullinane Corporation. Plus D/A: A Direct Access Software Security and Control System. Boston: Cullinane Corporation, 1972.

Fogg, L. W. CLOT: Card Library on Tape. Wilmington: International Telecontrol Corporation, 1970.

Interactive Sciences Corporation. TECO (Text Editor and Corrector) Reference Manual. Braintree: Interactive Sciences Corporation, 1971.

IBM Cambridge Scientific Center. A Conversational Context-Directed Editor, No. 320-2041. Cambridge: IBM Cambridge Scientific Center, 1969.

Management Systems Corporation. Program/Manage: A Program to Maintain Source Program Modules. Dallas: Management Systems Corporation, 1972.

Pansophic Systems, Inc. Panvalet: The Direct Access Library System. Oak Brook: Pansophic Systems, Inc., 1972.

Sigma Software Company, CFMS: Card File Maintenance System. Guthrie: Sigma Software Company, 1972.

Sperry Rand Corporation. UNIVAC 1100 Series Operating System Programmer Reference. Sperry Rand Corporation, 1971.

Stanford University Computation Center. MVT Milten/Wylbur Operator's Guide. Stanford: Stanford University Computation Center, 1970.

Van Dam, A., and Rice, D. E. "On-Line Text Editing: A Survey," ACM Computing Surveys, III (September, 1971), 93-114.

APPENDIX G

DATA DESCRIPTION IN FORTRAN, COBOL AND PL/1

AUTOMATIC SYSTEM FOR COMPUTER PROGRAM DOCUMENTATION

Working Paper No. 6

July 14, 1972

Data Description in Programming Languages
(FORTRAN, COBOL, PL/I)

by

Darrell Ward

Texas A&M University

Texas Engineering Experiment Station

6-1-a

ABSTRACT

The various forms of data encountered in FORTRAN, COBOL, and PL/I, are described at the local and global levels, along with ambiguities that arise within each language.

Suggestion for a syntax directed general purpose algorithm is also discussed.

TABLE OF CONTENT

, 00

FORTRAN

Local Level - Detailed

The IBM FORTRAN H compiler presents a local map of names encountered during a compilation. All names used as variables, statement functions, subprograms and internal functions.

All names are accompanied by descriptive tags. A brief description of each tag follows:

Tag	Description
A	The variable was used as an argument.
F	The variable has appeared on the right of an equal sign.
S	The variable has appeared to the left of an equal sign.
C	The variable is in COMMON.
E	The variable has appeared in an EQUIVALENCE statement.
IF	Indicates an internal function.
NR	The variable has not been referred to.
SF	Arithmetic statement functions.
XF	A subprogram.
XR	Variables, arrays or subprograms that are referenced by name.

The FORTRAN (H) compiler also produces a relative address for those names that are used internally. All functions and subprograms are assigned a relative address of 000000.

The type and length of each variable is also displayed as part

of the MAP option. For example, a double precision real variable would have a TYPE of R*8.

A description of COMMON blocks is also given. The name of each block is presented along with the length of the block. All variables in the block are described in the same manner as presented under the MAP option, with the exception of an identifying tag.

Following this data layout is a map of any equivalence made for the block. This is in the form of the variable name and its offset from the beginning of the block.

Local Level - Detail Suppressed

There is currently no method for suppressing detail. The FORTRAN (G) compiler does separate the scaler variables from the array variables. It seems a combination of this feature along with the COMMON way presented by the FORTRAN (H) compiler would be a good approach to a local level description of the data.

A restriction of the above to arrays only would also appear to be an approach to suppression of detail at the local level. The data layout for all arrays and any equivalences on an array would be quite functional at the local level. This could be accomplished by utilizing the source code and generating the data layout from all specification statements which allow dimensioning. Equivalences could be resolved before the final output of the layout.

Some type of abbreviated notation would be applicable in the presentation of the array structures. For example, the presentation of a 2-dimensional array with subscript limits of 5 and 10 respectively might be presented as follows

```
A(1, 1)  . . .  A(1, 10)
   .               .
   .               .
   .               .
A(5, 1)  . . .  A(5, 10)
```

A notation for describing the storage method used might also be considered. Consider the following:

```
A(1, 1)  . . .  A(1, 10)
   .               .
   .               .
   .               .
A(5, 1)         A(5, 10)
   *               *
   *               *
   *               *
```

1st 5 locations in memory for this array	last 5 locations in memory for this array

Global Level

There is currently no procedure for illustrating global data structures. A reasonable approach to this problem would require an analysis and collection of information in several categories.

The COMMON statement and any equivalences affecting COMMON must be considered for a global layout. The actual parameters in any FUNCTION or SUBROUTINE call must be analyzed since they are essentially global variables. An analysis of formal parameters in a SUBROUTINE or FUNCTION statement require analysis because they may certainly be representing a data structure at the global level. Actual parameters may also have an interaction with other variables via an EQUIVALENCE statement, thus the EQUIVALENCE statement must be inspected closely. Formal parameters may not be included in an EQUIVALENCE statement, hence we can disregard it in this instance.

The collection of the above information for subsequent processing is necessary for any global layout of data. Of course, it should be clear that relationships among the main program and subprograms will control the layout of any data relationships that exists across program boundaries.

It seems that a sophisticated system would be required to handle any data layout on the global level. Generally, the above problems can be applied to other high level programming languages such as COBOL and PL/I. Thus, a system should be designed so that it could be table driven and applicable to any of the mentioned languages. Table driven would essentially mean to supply to syntax and semantics of the few statements in each language which affects global layout.

Ambiguities

The EQUIVALENCE statement causes the sharing of locations in FORTRAN. This allows the partitioning and extension of one data structure by possibly several other data structures.

This presents problems at both the local and global level since COMMON variables and actual parameters may appear in the EQUIVALENCE list.

COBOL

LOCAL LEVEL-DETAILED

The use of the option DMAP will provide information about names in the COBOL source program. The following will be a description of the information that is output as a result of specifying the DMAP option.

The internal name generated by the compiler is output in order to facilitate the reading of the object code. The level number of the particular name is also generated when applicable. The description FD is used when the name is a file definition name. The source name used in the program is also output hence we have available three descriptions of importance: 1) the internal name, 2) the level number, and 3) the source name.

The remaining descriptors can be classed in the area of usage and location. All data names will have a base and displacement associated for the purpose of describing relative locations within the data structure. DECB and DCB information is presented in the case of file names. The storage assigned to the data name is described in terms of bytes used and also in terms of the type of data associated with that name. For example, the storage definition could be described in terms of characters or fullwords depending on the usage clause for that variable. This information is presented in assembler language terminology.

The usage of names is described in the following manner. For FD entries the access method utilized is identified. All group items are identified as such and all elementary items are described in terms of their USAGE clause.

All data - names that redefine other data names are described with the tag R. Any data names for which an OCCURS has been specified is described with the tag O. A Q indicates that the data - name is the object or contains the object of the DEPENDING ON option of the OCCURS clause. An M indicates that the format of the records of the file is:

F = fixed

V - variable

U - undefined

S = spanned

LOCAL LEVEL - DETAIL SUPPRESSED

Currently there is no procedure for suppressing the detail at the local level. The approach to pursue would parallel the FORTRAN and PL/I approach.

Two types of information would be informative at the local level in a functional sense. The data names which have OCCURS clauses specified for them are candidates for a data lay out. Again, as in FORTRAN and PL/I one must consider the possibility of memory sharing through the use of the REDEFINES clause. The second type of data layout would be created from the FD sections. Hence the record layouts could be described. This is essentially done vy the DMAP option yet many other variables are interspensed and the record layouts are not readily discerned in this environment.

The above approach would require an analysis of the DATA division quite rigorously. Again, as in FORTRAN and PL/I, a syntax driver algorithm could possibly be utilized for this purpose.

GLOBAL LEVEL

The conventions used in COBOL for subroutine linkages are quite similar to that used in FORTRAN. The key to a global analysis of COBOL would be the detection of the USING clause in the CALL statement. The isolation of all identifiers that follow and are non-scolar is the next step to be taken in the construction of a data layout.

The utilization of the source code as input and the output of the variables that are global type variables is the required task in such an environment. The consideration of the DEFINED clause is quite necessary to resolve or analyze any sharing of locations by variables.

AMBIGUITIES

The DEFINED statement is the COBOL statement that causes the sharing of locations in COBOL. The problem requires analysis in order to adequately describe both local and global layouts.

PL/I

Local Level - Detailed

The IBM PL/I compiler (F) produces a detailed description of all identifiers that are used in a PL/I program. The IBM compiler will be used as the focal point in the description of the information which can be obtained about identifiers. To activate a detailed printout of the attributes of identifiers, the user must utilize the compiler option ATR. The following description demonstrates the attributes that are presented at the detailed level.

All arithmetic variables have a set of attributes that may include the following:

a. BINARY|DECIMAL

b. FIXED|FLOAT (SINGLE|DOUBLE)

c. PICTURE

d. INITIAL

e. DEFINED

f. ALIGNED|UNALIGNED

g. dimension information

h. precision

i. STATIC|AUTOMATIC

j. INTERNAL|EXTERNAL

String variables are described with many of the same attributes as arithmetic variables. For example i) represents the storage class and j) represents the scope of the variable or identifier. These attributes are present for all types of identifiers. The following is a list of

attributes for string variables:

a. BIT|CHARACTER

b. PICTURE

c. INITIAL

d. DEFINED

e. ALIGNED|UNALIGNED

f. precision

g. dimension information

h. storage class

i. scope

One can easily see that many attributes are available via the ATR option. These are presented in alphabetical order with the declaration statement # associated with the identifier.

In addition to the above attributes, all elementary items of a structure are described. All major and minor structure names are correctly identified in the listing.

Other fringe type attributes are also available. For example, ENTRY, RETURNS, GENERIC, BUILTIN, etc. are descriptions of variables. Another area which may be considered is that of file descriptions. The ATR option permits the description of files. For example, such attributes as INPUT, OUTPUT, UPDATE, SEQUENTIAL, DIRECT are available.

The above demonstrates that a very detailed description of identifiers is available as output of the PL/I compiler. This is restricted to a single compilation and hence it applies essentially on the local level.

Local Level - Detail Suppressed

In addition to the attributes of the identifiers, the compiler produces an aggregate length table which is the length in bytes of all major structures and non structured arrays. This table is produced by the IBM compiler when the ATR option is used. This presents a very functional presentation of the overall structure of the major components of the identifiers.

Global Level

There are no present tools available to present a data layout over separate compilations. There are possibly two approaches to such a global description of data structures.

The first approach would necessitate a program to analyze the declaration statements of the source language. All identifiers which have the attributes of EXTERNAL and STATIC are candidates for a global data layout.

The second approach would require an analyzation and interpretation of the ESD which the compiler produces. All external symbols which are placed in a COMMON control section should be considered for a global presentation. In addition to these identifiers, under certain conditions control sections of the type SD will be created for global identifiers. This control section is not a COMMON control section but is accessible by other compilations thus, in effect, it is common to other procedures.

Ambiguities

The possibility of misinterpreting or misrepresenting the data layout exists because of the DEFINED attribute. The DEFINED attribute essentially causes the sharing of memory by several identifiers. This presents a problem in the presentation of the data layout since we may now have several arrays and structures utilizing the same memory locations.

This problem is also complicated by allowing character arrays to have other names specified for part of the character strings. This is implemented by the use of the POSITION attribute in conjunction with the DEFINED attribute.

CONCLUSIONS

Clearly, if one considers the source languages of the three languages at a global level, it becomes apparent that all have similar methods for transmission of variables, redefinition of variables and declaration of arrays and structures. PL/I is the encompassing language in the sense that it includes the features of both COBOL and FORTRAN.

Since the features, which must be considered, are quite similar the approach of a syntax driven collection algorithm might be quite feasible. The fact that considerably enhances such an approach is that fact that such a limited analysis of source statements is necessary. The main considerations are to be given to specification statements and calling - answering sequences. The types of specification statements will include array specification, structure specification and redefinition statements.

The above approach could be applied for a global layout and a subset could be utilized at the local level for a functional picture of the data structures. The local level detailed layouts are handled adequately by the compilers and the emphasis should be placed on the detail suppressed mode and global description.

APPENDIX H

DOCUMENTATION STANDARDS

AUTOMATIC SYSTEM FOR COMPUTER PROGRAM DOCUMENTATION

Working Paper No.4

July 14, 1972

Documentation Standards

By

Charles Schroeder

Texas A&M University

Texas Engineering Experiment Station

11-12

ABSTRACT

This paper investigates the state of the art of documentation standards. The few standards that do exist are reviewed and standards at various levels are suggested where possible. It was found that existing automatic documentation packages fail to meet any sort of standards.

Systems Documentation and Program Documentation are then discussed with an overview of what is generally acceptable and what might be incorporated into a set of standards.

Existing flowchart and decision table standards are presented.

TABLE OF CONTENTS

INTRODUCTION

Basically, Systems Documentation may be categorized into developmental documentation and control documentation. Since one of the major problems constantly discussed in almost any technical or administrative setting is the problem of poor communication, the above documentation types are intended to lessen this problem. Developmental documentation is descriptive of the system itself, i.e., performance characteristics, tools and materials. It is therefore the means of communication <u>about</u> the system. Control documentation is concerned with communicating information about resources used to develop the system and involves project development organization, personnel, time, materials, and money.

Systems designed are never static. They must change to meet new customer requirements or in order to be compatible with hardware and software modifications. More often than not the designer of the original system is no longer available to make the necessary changes, therefore a clear and complete record of what each system does and how it was developed is essential. Hence, at least some standard must be developed in order to assist the programmer/analyst in preparing good documentation to establish a common language of communication.

Program documentation deals more specifically with producing what might be called a Job Documentation Package or a Program Manual. In general, the aim in writing programs is to use the computer to solve problems which are too tedious, repetitive or uneconomical for manual methods. From some

problem definition and solution specification, detailed program logic is designed, and the program coded and tested. The product is thus a proved and documented (hopefully) program ready for operation, probably initially operated within a test environment.

There is a wide range of application differences, however within such a definition of a program, from a scientific problem-solving application which can perhaps be defined and solved by a single person to a detailed systems specification requiring many designers and programmers.

The former may serve a one-time purpose and specific records of input and file records may not be necessary. In this case rather little documentation may be required.

The case of the detailed systems specification made up of a number of program specifications describing some complex data processing system is much more complex. Here, many "in-house" programmers will be involved and possibly outside contractors will be employed. In this case, it is generally agreed that accurate program documentation is a must.

The real problem in setting standards is to specify how much documentation is required and from whom. No national standards for program documentation exist, but some software packages are commercially available to produce automatic documentation at the program level. Although rather poor in general and far from meeting even the most loosely defined set of standards, they are an attempt at standardization.

SYSTEMS DOCUMENTATION STANDARDS

State of the Art

In any discussion concerning standards, it is first necessary to lay some ground rules. First of all, what are standards and what has been standardized?

Currently the only set standards for computer software and/or computer software documentation deal specifically with standard flowchart symbols and abbreviations list, and decision table standards. Beyond these items, the word "standards" generally refer to what a particular installation feels is necessary. Very little information has been found concerning standards of any kind, but most reports on any type of standards do agree in some broad areas.

At present, there are no universal documentation standards which are directly applicable to all installations. This is true because of the type and level of complexity of documentation for one location may be inadequate or inappropriate in another environment. Each data processing department must therefore implement a documentation system which suits its own environment. This could be done by adapting a general documentation system to local conditions.

The development of a documentation system for a particular data processing organization must take account of the major influencing factors which include:

(1) management commitment

(2) project characteristics

(3) corporate environment and organizational characteristics, and

(4) technical environment

In any event, with respect to systems documentation, the following documents <u>might</u> be recommended.

(1) A User Request - which is the initial approach for the user for data processing assistance. It contains a brief problem description.

(2) The Systems Proposal - essentially a major report resulting from a systems survey. It may be considered as a feasibility report, including a detailed specification of the recommended approach and project plan.

(3) The Analytical Report - essentially a project plan, supplemented as necessary with evaluation information.

(4) The Design Requirement Statement - which is essentially a specification of requirements in systems terms.

(5) The Systems Summary - is a general description of the complete system or system change.

(6) File Specification - a detailed description of the purpose, contents and organization of a file.

(7) Transaction (Input) Specifications - This document describes all inputs to the system.

(8) Output Specifications - This document details the systems outputs- why and when produced, contents, formats and recipients.

(9) Segment (Processing) Specification - essentially a statement of the design requirements and general logic for a program.

(10) Systems Test Plan - A permanent record of the testing procedure to prove the system prepared by the systems design function.

(11) Programming Specifications - (which are discussed separately later)

The above list of systems documents are basically those of Gray & London in Documentation Standards.[2]

The present state of the art on the above set of documents follows the trend of filling out appropriate printed forms for each step in the documentation process. These forms are generally suited to the individual needs of the specific installation and seem to follow no real fixed pattern. No mention is made of automatically producing any of these forms or their equivalents. It seens very feasible that a great deal of this documentation could be automatically produced, as is discussed in other reports.

Suggested Standards for Systems Documentation

The problem of how much documentation is needed for a particular system is the obvious problem here. In addition to the standards mentioned in discussing the state of the art, minimal standards should include most of the following.

I. Systems Flowcharts - Flowcharts provide a means of identifying programming functions and graphically visualizing the logic and the path of its flow in the solution of a problem. Generally, two levels of flowcharts are recommended. A third level, a MICRO or logic detailed flowchart may be desirable but can be considered as optional.

A. First Level System Flow. This level should be a general system flow, giving an overview of the major processing areas and programming requirements (programs) that will be required for system development. This flow should be in block form to identify the core resident programs and the programming path necessary to accept, process and output data. The general flow should also show the support routines, e.g., overlay, and common routines. The first level flow may require more than one page to identify core resident and support programs or routine. Core resident programs should be grouped within a block to show the processing area to which the program belongs. Systems should be designed so core resident and support routines/programs are developed in modular form.

1. Core resident and support routines developed to aid processing and message switching functions should be identified by name or subject, followed by some distinguishable alphanumeric program name.

2. A flow depicting the hardware configuration should be drawn showing all hardware and peripheral components, and the path of the data flow through the system.

B. Second level MACRO flow. The programming functions identified in the first level system flowchart should be developed into MACRO flow (semi-detailed) diagrams.

1. Some method for identifying each symbol or block of the MACRO flow should be used.
2. The MACRO flowchart expands the program or major processing function of the first level into semi-detailed component blocks or programming symbols. Flowchart pages should be numbered.
3. As the second level flowcharts are reviewed by the lead programmer/analyst, processing errors and omissions should be detected prior to coding.
4. The MACRO flow should depict, in graphic and symbolic form, input/output functions, subroutine, major processing functions and the processing sequence for the coding of a program. It should be in sufficient detail to permit another programmer to develop the source coding.
5. Standard flowcharting symbols should be used.
6. The general flow should be from top to bottom, left to right.
7. Off-page connectors should contain unique tags or

coordinates to which they point, and the flowchart page number. If the off-page connector does not relate to the preceding page, show the page number of the page where the flow was broken, e.g. from page 4 of 12.

8. On-page connectors should contain unique tags or coordinates to which they point.

II. Narrative Documentation

A. Systems Reference Manual.

While the system is under development, a general narrative should be written giving a brief history and purpose of the system. This will serve as the composite document for the system and will be developed concurrently with the system. The introductory chapter will give the development history and identify each program/routine and contain a brief statement giving the purpose and function of each program/routine. This first chapter should contain the first level block diagram flows. Subsequent chapters should contain the second level MACRO flowchart and a general narrative for each program.

B. Operator (User) and Test and Acceptance Manuals Information for the Operator's and the Test and Acceptance Manual, as applicable, can be lifted from the program abstract and other narratives as the system is being developed. This information, along with the Systems Reference Manual, will be developed concurrently with the system.

C. When a system is ready for implementation, system documentation should be submitted to a Documentation Library.

III. "Stand-Alone" Programs or Modifications to existing programs.

A. When a "stand-alone" program is developed or a modification to existing programs takes place, it should be narratively documented.

B. When the program is ready for release, the documentation, as required, should go to some Documentation Library.

PROGRAM DOCUMENTATION STANDARDS

State of the Art

Just as there are many different ways to write a program as there are programmers, there are as many ways to document a program. To determine what program documentation is necessary or sufficient is the purpose of program documentation standards.

The United States Air Force[14] feels that acquiring and maintaining accurate and up-to-date documentation is an essential part of the software production and maintenance process. Well-documented programs are necessary in effective communication of software system ideas and techniques between organizations with both operational and economic benefits. In addition, good documentation

1. allows the lead programmer/analyst to review the efforts of an individual programmer to ensure conformance with system design criteria.
2. assists programmers in testing and debugging.
3. enables programmers to update old programs with minimal difficulties.
4. permits new programmers to learn their systems without having to recode programs to discover what they do.
5. makes available information that can be used in the design of new systems within a given unit and to eliminate duplication of effort wherever possible.

Recommended Standards for Program Documentation

The following is a set of general recommendations which can be adopted, or at least partially so, as the required standards for program documentation. These standards were collected from several sources, each different installation making suggestions with slight variations.

A consensus of the standards reviewed recommends that a program manual, a complete final document of a program, should be prepared. It should contain:

1. a general description of the function, use, and methodology of the program.
2. a description of input, files and output used or produced by the program.
3. flow diagrams showing the logic of the program. (Flow diagrams are discussed later in this report.)
4. a description of instructive output messages, e.g. output on console or printer, etc.
5. coding information, e.g. an assembly listing, memory print, descriptions of matrices or tables used.
6. a test plan.
7. program test and operating instructions.

Operations Documentation should be prepared partially by the system designer and partially by the programmer. This documentation, prepared for both the user and the data processing operations staff, should contain:

1. Program Test Instructions, a document comprised of those instructions which are necessary to guide the computer operator in running a test program, prepared by the programmer.
2. Systems Operating Instructions, a list of processing steps, in execution sequence, defining all operating requirements. It should include:
   a) Summary workflow schedule
   b) Data collection and preparation instructions
   c) Input control instructions
   d) Job assembly instructions
   e) Output review and control instructions

   A basic collection of forms for System Operating Instructions would include:
   a) workflow summary
   b) general clerical
   c) data preparation (i.e. keypunching)
   d) auxiliary machine (by machine category)
   e) computer operating (by computer type)

   Each operating instruction form should bear the basic identifying information comprised of at least:
   a) systems identification
   b) operation identification (brief title)

c) step number for this operation

d) previous operation step number/next operation step number

e) date

f) originator/authority

Any number of other entries may appear on each form. These may include:

a) responsibility for performing the instructions

b) input: description and source

c) output: description and destination

d) process: summary of functions performed

Each specific type of form will have further detail pertinent to that particular form. These forms can be quite lengthy and, rather than list the necessary information for each, a few examples follow on the next pages.

The forms featured in the above suggested documentation can vary greatly. Several different forms (i.e. printed forms) may be required for any one of them. These forms can be detailed or general, depending upon the particular user's needs. Military Standards[6] generally require detailed information while smaller systems, with little outside communication necessary, could be documented adequately with less detail.

At present, most program documentation information is recorded by hand. It would be desirable if this documentation could be generated automatically from the program and from previously prepared systems documentation. No present documentation package offers any means for

doing this. However, it would be relatively simple to require the programmer to input any additional underivable information using a CRT. These forms could be produced on the screen and the programmer would enter the necessary information. This information could then be printed with the program output.

Flowcharts can be produced by some packages now available. Flowcharts are in a later section of this report.

Sample Program Documentation Forms

MACHINE SET UP FORM

PROG NAME______________ PROG # ______________ USER I.D. CODE______________
AREA ______________ PROJECT ______________ PROGRAMMER ______________
JOB PREQ ______________ REGION SIZE ______________K

DIRECT ACCESS REQUIREMENTS:

PERMANENT

USER ASSIGNED PACK	DATA SET NAME	DDNAME	SERIAL # / CELL #	BIN #

TEMPORARY

USER ASSIGNED PACK	DATA SET NAME	DDNAME	SERIAL # / CELL #	SPACE =

TAPE REQUIREMENTS:
#7 TRK. UNITS ______________ # 9 TRK. UNITS ______________

DATA SET NAME	DDNAME	LABEL TYPE	7 OR 9 TRK	DEN/ MODE	WRITE RING IN	RET PD IN DAYS	OUTPUT FROM RUN #	INPUT TO RUN #

(FOR ADDITIONAL INFORMATION ATTACH ANOTHER SHEET)

ADDITION ☐ REPLACEMENT ☐

DATE _____ SECTION _____ PAGE _____

CARD READER REQUIREMENTS:

DATA SET NAME	DDNAME	SOURCE	DISPOSITION

CARD PUNCH REQUIREMENTS:

DATA SET NAME	DDNAME	POCKET #	DISPOSITION

PRINTER REQUIREMENTS:

DATA SET NAME	DDNAME	PRINT TRAIN	FORM #	SETUP #	LINES PER INCH	BURST	DECOL-LATE	DISPOSITION

ADDITION ☐ REPLACEMENT ☐

DATE _____ SECTION _____ PAGE _____

DOCUMENTATION APPROVAL FORM

Check Type of Release: ☐ NEW PROGRAM ☐ MAINTENANCE ☐ SYSTEM

SYSTEM OR JOB NAME	PROGRAM #	PROGRAMMING AREA	DIRECT INQUIRIES TO:
NAME OF PROJECT	NAME OF PROGRAMMER		DATE

DESCRIPTION OF SYSTEM, NEW PROGRAM OR MAINTENANCE:

Program Configuration:

DIRECT ACCESS STORAGE
☐ Permanent ☐ Temporary

DIRECT ACCESS DATA SETS
☐ Tables ☐ Core Res. Modules
☐ Disk Resident Modules
☐ Data Sets

Items Affected By This Release:
☐ Program ☐ Documentation
☐ Tables or Data Sets
☐ External Procedures

Maintenance Priority:
☐ EMERGENCY-Contact Name: ______
Phone: ______
☐ SPECIFIC DATE ______ CYCLE ______
☐ WHEN POSSIBLE

SUBMITTED BY

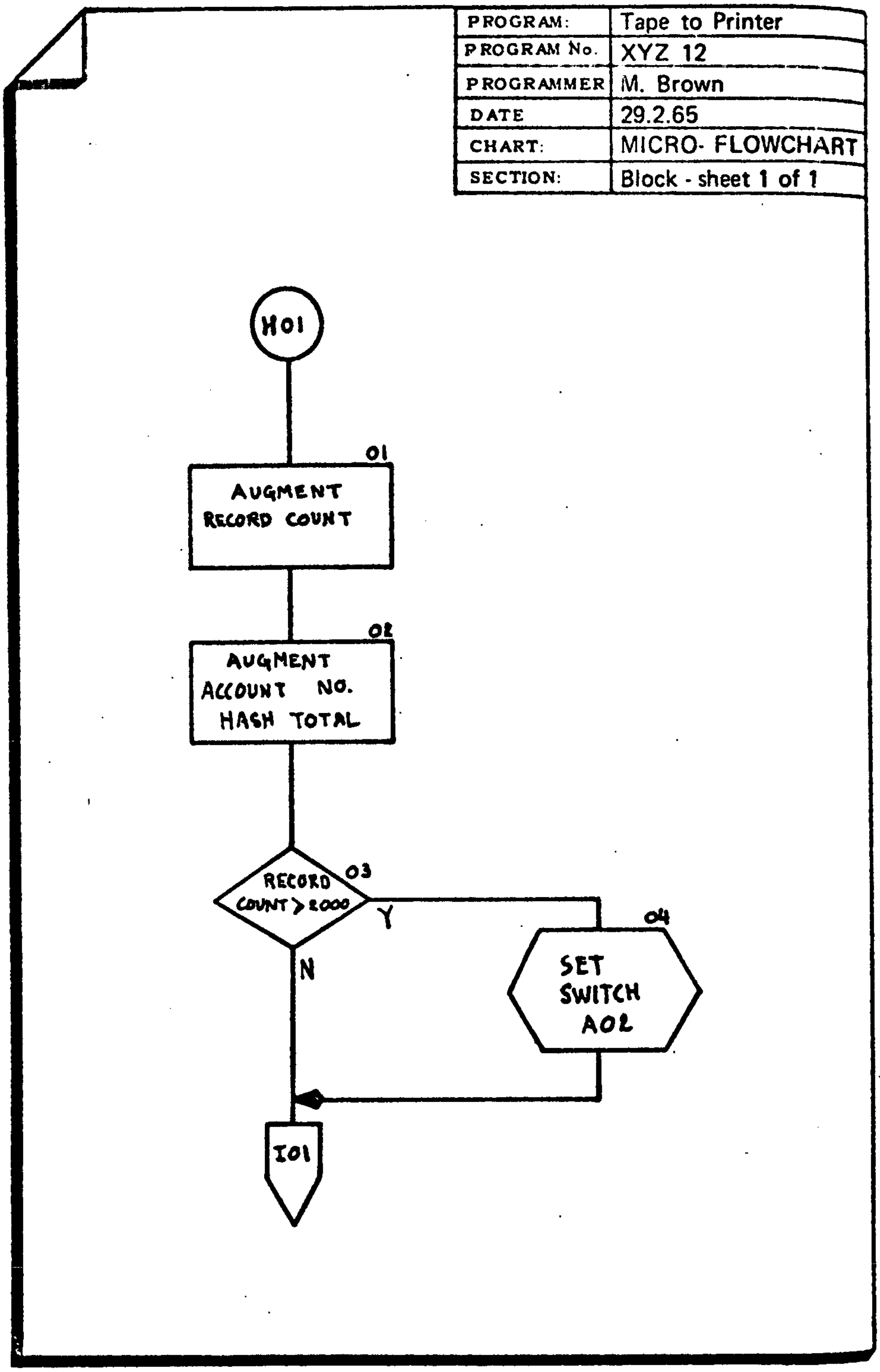
Complete This Section For New Program and When Maintenance Affects One of the Following:

PROGRAM INFORMATION - CHECK ONE FOR EACH ITEM:
JOB TYPE: ☐ Batch ☐ Teleprocessing
RUN FREQUENCY: ☐ Daily ☐ Weekly
☐ Monthly ☐ Quarterly
☐ Annually ☐ On Request
☐ One Time ☐ ______
ESTIMATED TIME: ___ Minutes
LANGUAGE: ☐ COBOL ☐ FORTRAN
☐ RPG ☐ S/360 AL
☐ PL/1 ☐ ______
CORE STORAGE: Enter Region Size ______K
(Do Not Include OS)

EQUIPMENT INFORMATION - ENTER WHERE APPLICABLE:
☐ 360/65 ☐ 1401
☐ Card Reader ☐ Card Punch
☐ Printer ☐ Drum
☐ Optical Scanner ☐ Data Communications
___ Number of Tape Units ___ Number of Disk Storage Drives
___ Number of Data Cell Drives ___ Number of ______

APPROVALS (Manager Level or Higher)

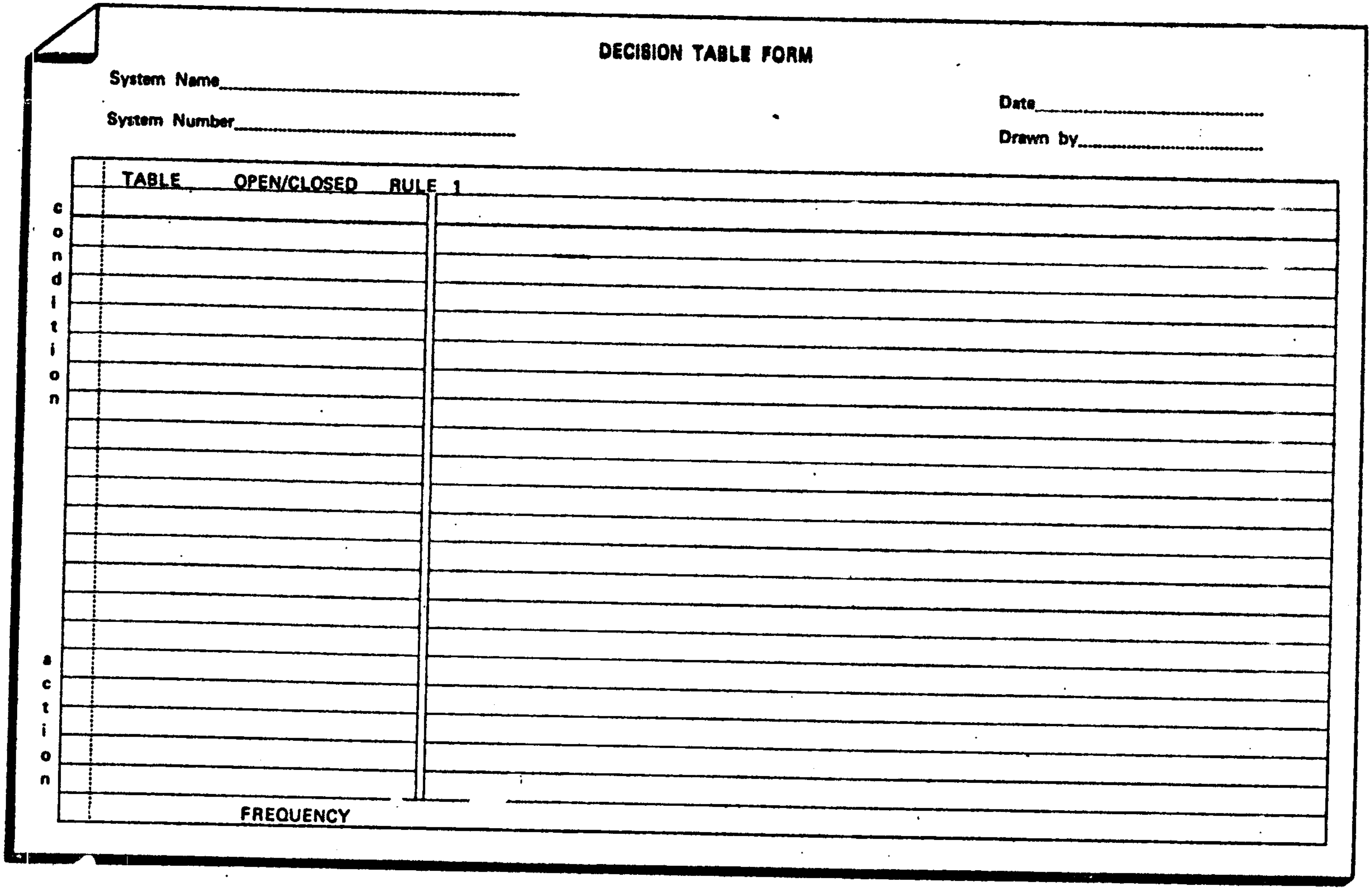
	NAME	DATE
(Compl. this line for New Program Only) SUBMITTING OFFICE: PROGRAMMING AREA:		
(Compl. this line for Maintenance Only) MAINTAINING OFFICE: PROGRAMMING AREA:		
MANAGER REVIEW		
REASON FOR REJECTION OR COMMENTS:		



Sample Micro-flowchart

SYSTEM: Stores Stock (1) DATE: 5-6-67 BY: J.M. Dumf

Process Step No.	Operation (Process) Flowchart	Specification No..	Volumes	Latest Time To This Operation	Latest Time Out From This Operation
1	Check receipt of batches from all stores	1.1	17 batches of approx. 900 to 1200 documents	10.00 Tuesday	12.00 Tuesday
2	Punch/Verify Issue/Receipts	1.2	17,000 cards	12.00 Tuesday	17.00 Thursday
3	Balance Batches	1.3	N/A	17.15 Thursday	19.00 Thursday
	Agreement (N / Y)				
4	Error Check to Source	2.0		Variable	
5	Prepare Monthly Parameter Cards	1.4	N/A	08.45 Friday	09.00 Friday
6	Take-on Program 'JM37'	1.5	17,000 cards 120 forms	09.00 Friday	11.45 Friday
	2				



DECISION TABLE FORM

System Name ______________________

System Number ______________________

Date ______________________

Drawn by ______________________

TABLE	OPEN/CLOSED	RULE 1

condition

action

FREQUENCY

Sample decision table form

DECISION TABLE FORM

System Name Airline Reservation Example

System Number CX

Date September 1967

Drawn by R. Tomms

		TABLE L3 OPEN/CLOSED RULE	1	2	3	4	5	6	7
condition	1	First class requested	Y	Y	Y	Y	-	-	-
	2	Tourist class requested	-	-	-	-	Y	Y	Y
	3	First class available	Y	N	N	N	-	Y	-
	4	Tourist class available	-	Y	-	N	Y	N	N
	5	Alternate class acceptable	-	Y	N	-	-	Y	N
action	6	Write first class ticket	X	I	I	I	I	X	I
	7	Write tourist class ticket	I	X	I	I	X	I	I
	8	Adjust 1st cl. seat invent.	X	I	I	I	I	X	I
	9	Adjust tourist seat invent.	I	X	I	I	X	I	I
	10	Suggest another flight	I	I	X	X	I	I	X
	11	Go to next request	X	X	I	I	X	X	I
		FREQUENCY	20	8	6	10	40	9	7

Sample limited entry decision table

DECISION TABLE FORM

System Name Airline Seat Reservation

System Number CX

Date September 1967

Drawn by R. Tomms

		TABLE A2 OPEN/~~CLOSED~~ RULE	1	2	3	4	5	6	7	8	9	10
condition	1	Request for	First	First	First	First	First	Tourist	Tourist	Tourist	Tourist	Tourist
	2	Space available	First	Both	Tourist	Tourist	None	Tourist	Both	First	First	None
	3	Alternate class	-	-	OK	NG	-	-	-	OK	NG	-
action	4	Reduce seat inventory	First	First	Tourist	-	-	Tourist	Tourist	First	-	-
	5	Write ticket	First	First	Tourist	-	-	Tourist	Tourist	First	-	-
	6	Go to next	Req	Req	Req	Flight	Flight	Req	Req	Req	Flight	Flight
		FREQUENCY										

Sample extended entry decision table

NATIONAL STANDARDS

Flow Chart Standards

The ANSI Standards define three major groups of flowchart outlines.

1) Basic - this includes specifications of symbols representing four functions considered to be the minimum symbols required for adequately representing data processing action. These functions are input/output, processing, flow direction, and annotation.
2) Specialized - consists of outlines for specifying three distinct groups:
   a. data - carrying media (document, magnetic tape, etc)
   b. peripheral equipment type (on-line storage, menual input, etc.)
   c. selected types of processing action (decision, sort, collate, etc.)
3) Additional - these outlines include symbols for representing origins, terminations and functions.

It appears that flowcharting symbols are one of the few things that are generally accepted as standard. These standards are included on the following pages.

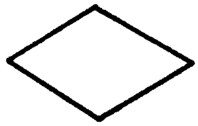
Decision Tables

Some standards have also been set up for use with respect to decision tables. These standards generally adhere to the rules which follow. Examples are also provided.

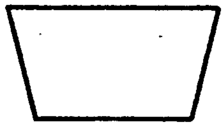
SYSTEMS FLOWCHART SYMBOLS

1. Process Symbol used to represent any kind of processing function, or any operation for which no particular symbol is provided.

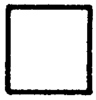
2. Decision Symbol used to represent a decision that determines which of a number of alternative paths is to be followed.



3. Manual Operation Symbol



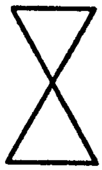
4. Auxiliary Operation Symbol



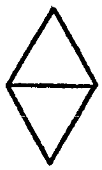
5. Merge

6. Extract

7. Collate



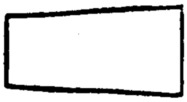
8. Sort



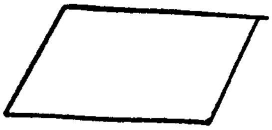
Standard symbols for system flowcharts

SYSTEMS FLOWCHART SYMBOLS

9. Manual Input



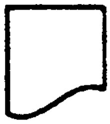
10. Generalized Input/Output Symbol



11. On-Line Storage Symbol represents the use of any kind of on-line backing, store, i.e., disc, drum or magnetic tape

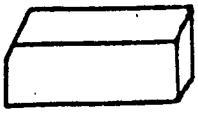
12. Off-Line Storage Symbol represents the function of storing information off-line, regardless of the medium on which the data is recorded.

13. Document

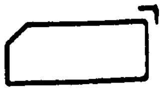


14. Punched Card

15. Deck of Cards

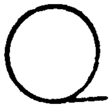


16. File of Cards: this symbol represents a collection of related punched card records.



17. Punched Tape

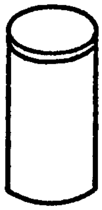
18. Magnetic Tape



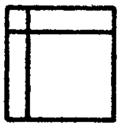
SYSTEMS FLOWCHART SYMBOLS

19. Magnetic Drum

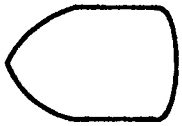
20. Magnetic Disc



21. Core Store



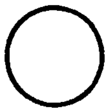
22. Display



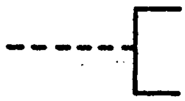
23. Communication Link: this symbol represents transfer of information by a telecommunication link.

24. Graph Plotter

25. Connector



26. Comment



27. Flow Indicators

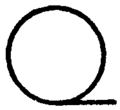
PROGRAM FLOWCHART SYMBOLS

1. General Operation Symbol: used for any operation which creates, alters, transfers or erases data, or any other operation for which no specific symbol has been defined in the Standard.

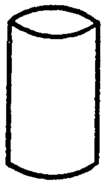
2. Subroutine (Predefined Process) Symbol: used when a section of program is considered as a single operation for the purpose of this flowchart.

3. Generalized Input/Output Symbol: used where it is desired to stress I/O operations. The symbol is used as an alternative to the specific device symbols when:
   – at the time of flowcharting the actual device to be used has not been decided,
   – the flowchart is drawn as an example, and is not related to any specific I/O function,
   – local standards specify its use.

4. Magnetic Tape I/O

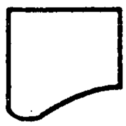


5. Disc I/O

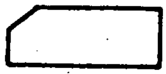


6. Drum I/O

7. Document I/O



8. Punched Card I/O



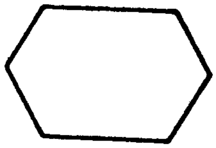
9. Punched Paper Tape I/O

Program flowcharting symbols

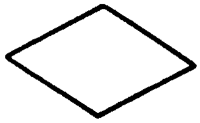
C. 5

PROGRAM FLOWCHART SYMBOLS

10. Preparation Symbol: used where it is desired to accentuate an operation that partially or completely determines the selection of a particular exit at given Branch Symbols.

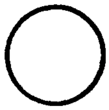


11. Branch Symbol: has one entry line and more than one exit. The symbol contains a description of the test on which the selection of an exit is based. The various possible results of this test are shown against the corresponding exits.

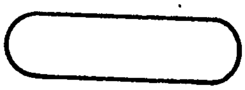


12. Offpage Connector Symbol: used as a linkage between two blocks of logic that are to be found on separate pages of the flowchart. The symbol is only used on the 'exit' page, on the 'entry' page an onpage symbol is used.

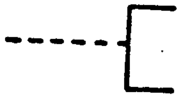
13. Onpage Connector Symbol: used as a linkage between two blocks on the same page, when it is not desirable to connect them using a linkage line. The label of the block to which the connection is being made is written inside the symbols.



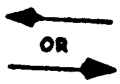
14. Terminal Symbol: used as the beginning or end of a flowline (e.g., start or end of a program).



15. Annotation Symbol: used to add additional information to a symbol or block of program.



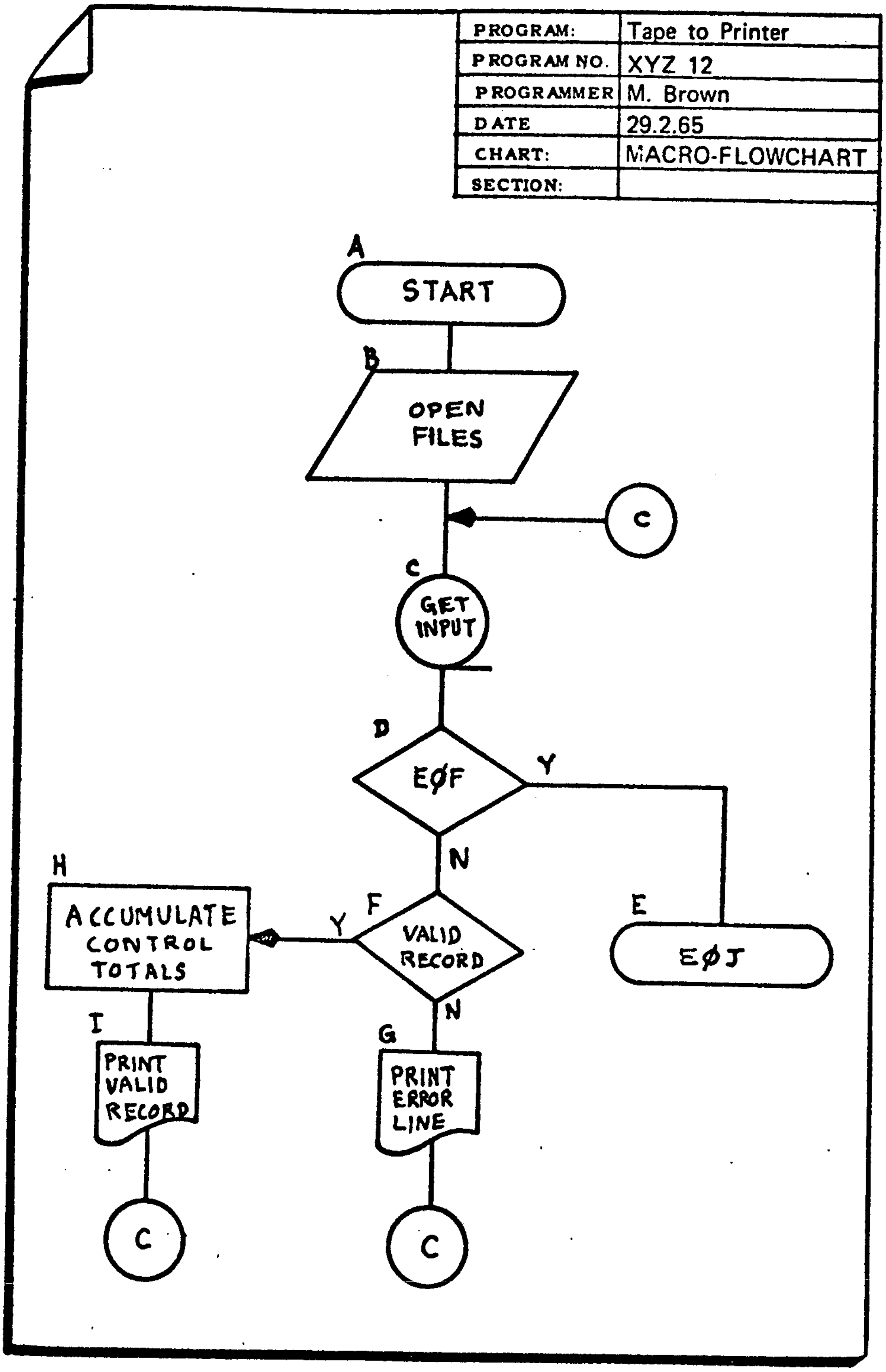
16. Flowlines (Linkage Lines): used to show the flow between blocks of a flowchart. The normal flow is from top to bottom and left to right of the page. The programmer may dispense with the use of the direction arrows when the chart follows the normal flow. They must be used, however, for any portion of the diagram which does not follow the normal flow.



List of Standard Symbols and Abbreviations

+	Plus or positive
–	Minus or negative
±	Plus or minus, positive or negative
x	Multiplied by
÷ or /	Divided by
=	Equals
≠	Does not equal
>	Greater than
<	Less than
≥	Greater than or equal to
≤	Less than or equal to
c(x)	Contents of location X
cf or :	Compare or compared with
→	Used within an operational symbol to denote transfer of data
EØF	End of file
EØR	End of reel
EØJ	End of job
#	Reserved for local use
No.	Number

Standard symbols and abbreviations list



Sample Macro-flowchart

These rules are:

1) Decision tables should be drawn on the form shown at the end of this section. The layout of the elements of the decision table should be:

CONDITION STUB	CONDITION ENTRY
ACTION STUB	ACTION ENTRY

The double line separating the stubs from the entries is predrawn. The designer should draw his own double line to separate condition and action areas of the form.

2) Only one table should be drawn on each sheet.

3) Tables must be named at the head of the table. A name of the form "TABLE XX" is preferred, but other naming standards may be specified locally. Following the name, the words OPEN or CLOSED should be written to indicate they type of table.

4) No decision table should be drawn that has:

   a. more than 4 condition variables, if neither dashes nor an ELSE rule are used, or

   b. more than 6 condition variables if dashes and/or an ELSE rule are used;

   c. more than 12 decision rules, and

   d. more than 15 action variables.

5) Blanks must not be left in the condition entries. Dashes should be used to indicate that the value of a condition does not affect a particular action.
6) On the condition entries, Y should be used to indicate the truth and N the falsity of a condition.
7) On the action entries, X should be used to indicate that an action is to be followed and I to indicate that it is to be ignored.
8) Actions must be written in the order in which they are to be executed.
9) Every effort should be made to combine rules within a table which give rise to the same action. It will often be found that the value of one condition is immaterial.
10) Tables must be drawn up in such a way that all rules are true alternatives; rules may be examined in any order but only one rule can satisfy a given set of conditions.
11) The final action entry for each rule must specify where to go next.
12) Where the information is available, it can be of considerable assistance to the programmer if the expected frequency of satisfaction of each rule is indicated on the table.

DOCUMENTATION PROCEDURES CURRENTLY IN USE

The information in this section was selected from material requested from various companies about their respective documentation procedures. The replies received ranged from very brief and indefinite requirements to highly specialized and detailed specifications. In general, the size of the company and the amount of computer use seemed to affect the volume of documentation required. It should be noted that the standards mentioned below are a subset of the documentation standards recommended earlier in this report.

In general, the following items seem to make up some sort of accepted "standard."

1. Abstract - This was required by most of the companies involved. The detail of the abstract did vary a little, but the basic ideas seemed constant. This generally included items such as Program Title, Program Number, Language, Machine Configuration, I/O Description, etc.
2. Source Program Decks and Listings - There were some variations here. In general, source programs were kept on file and one company even suggested maintaining actual card decks if the source program was not on tape. In addition to this, general setup of Input, JCL, and special information about control of the program was also used.
3. Program Flowcharts in general were required as some part of a documented program. The level of these flowcharts varied. One

company specified the use of AUTO-FLOW for program logic flow.

4. Systems Flowcharts were used by some of the companies involved while others made no mention of them.
5. Almost all information received made mention of procedures for updating documentation for revised jobs. This process varied from a detailed set of steps of completely redocumenting the program to merely adding update information to the previous documentation.
6. Details of Data Layout was generally required. This included information about data cards, tapes and other information necessary for processing. Keypunch instruction was also requested at some levels.
7. An Operator's Guide was required by a few of the companies involved, however, this was not required by all.

The above list makes up the bulk of the required documentation. The detail required obviously varied greatly. Some major companies required considerably more than the above mentioned items, while a very small company included required only comments in the program listing.
It was interesting to note that in this latter case, 3 levels of comments were specified, depending upon the number of asterisks appended to the comment.

AN EXAMPLE OF DOCUMENTATION

One of the better examples of automatically produced program documentation which was found was from the USAF. This message switching program included in its computer printed output the following:

(1) A detailed table of contents, which included a list of major headings along with subheadings of everything included in this listing. Also included was notation used for separating sections, sub-sections and further subdivisions via lozenges, periods and asterisks.

(2) Section I of the output was called "1108 STANDARDS". This included a list of 23 requirements for running on the particular system used. This included information concerning external labels, entrance and exit requirements, tags, external drum equates, information about standard date, tape codes, etc.

(3) Section II was called "A CHRONOLOGICAL HISTORY OF COLUMNS 79 AND 80". This reference to the columns mentioned gave in detail information about successive assembly dates and implementation timetable. The listing included all changes made in these assemblies and their dates. These columns are used to record program modification levels. This section also gives a detailed explanation of what these modifications were and why they were made.

(4) Section III, "PROGRAM INTERFACE AND RELATED INFORMATION". This included instructions to follow in interfaces, overlays, flags and related information. Also included in this section is information on interrupt analysis.

(5) Section IV, called "BUFFERS, MAPS, AND TABLES", gave drum maps, circuit lists, and several pages of tables. Included was polling information, table assignment tables, and requirements for polling. All of this information was dated.

(6) Section V, called "PROGRAM DESCRIPTIONS", contained the names of and information about all program segments used in this run. Each program was listed separately and its contents and relation to other programs were given. Information concerning on-line or not, where control is received from, specific function of program, what all message codes mean, register usage, and meanings of switches and table indexes were included.

Following the above information, which, incidentally, covered 85 printed pages, came a complete program listing for each program used. Comments were used freely throughout.

It should be emphasized that all of this information (196 pages of it!) was printed output. Assembly dates and codes were included on each entry.

CONCLUDING COMMENTS

The preceding pages give a summary of what might be desired in some standards for systems and program documentation. The automatic production of such information is a more challenging problem. Not only do standards not exist, but to aggravate matters, documentation software now available fails to meet any of the existing standards. Flowcharting packages do not follow the "rules" already established and standard symbols are not consistently used.

There is no reason to believe that reasonably good automatic documentation cannot be developed. Two possible approaches to achieve this goal might be: first, if some reasonable documentation standards can be established, users should then force software vendors to meet these standards. Secondly, if a good automatic documentation system could be developed which would satisfy a large portion of the users, its general acceptance could make it the standard. The establishment of good standards for program documentation will enhance communication at the local project level, between different user groups and/or with outside contractors.

BIBLIOGRAPHY

1. Arseven, Susan, Texas A&M University DPC System Documentation Guide, 1972

2. Gray, Max and London, Keith R., Documentation Standards, Brandon/Systems Press, Princeton, 1969

3. IBM Study Organization Plan, IBM Corporation, 1963

The Approach	F20-8135
The Method Phase I	F20-8136
The Method Phase II	F20-8137
The Method Phase III	F20-8138
Documentation Techniques	C20-8175

4. Computer Program Documentation Guideline, NASA, 1971

5. Documentation Standards, USAF - Tinker AFB, 1971

6. Military Standards #480, #483, #490, USAF

7. MRI Systems Corporation, Samples of Documentation, (Not a publication)

8. Mouton, Melba L., Documentation: Motivation and Training or Automation, Lecture Notes presented at GSFC,1970

9. Advances in Computers, Vol. 8, Academic Press, New York, 1967

10. Naval Command Systems Support Activity, Documentation Specifications for Functional Programs, Technical Note, NAVCOSSACT Document Number 59T700 TN-02

11. ANS Flowchart Symbols and their Usage in Information Processing, American National Standards Institute, Inc., New York, 1970

12. Standards Guide for Space and Earth Sciences Computer Software, Goddard Space Flight Center, Maryland, January 1972

13. Program Documentation Standards, Bell Systems Practices, American Telephone and Telegraph Co., 1966

14. Automated Weather Network, Automated Weather Division, CCPC, Tinker Air Force Base, Oklahoma.

AUTOMATIC SYSTEM FOR COMPUTER PROGRAM DOCUMENTATION

Technical Proposal

Submitted to

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771

Submitted by

Data Processing Center
Texas Engineering Experiment Station
College of Engineering
Texas A&M University
College Station, Texas 77843

1-1a

October 1972

APPENDIX I

AUTOMATIC SYSTEM FOR COMPUTER PROGRAM DOCUMENTATION - A PROPOSAL FOR INITIAL IMPLEMENTATION

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

Part I of this proposal discusses the technical approach to the implementation of an automatic system for computer program documentation. Section 1.0 contains a statement of the philosophy. Section 2.0 contains general background information and a concise summary of studies that led to the system design. Section 3.0 contains detailed design specifications for the automatic documentation system. Section 4.0 lists the advantages of the initial phase of the implementation of the automatic documentation system and Section 5.0 covers future expansion. Plan of activity, reports and documents, program organization, personnel qualifications, facilities, and program schedule are contained in Sections 6 through 11.

Part I - Technical Proposal

1.0 INTRODUCTION

The Data Processing Center at Texas A&M University is pleased to submit this proposal to the National Aeronautics and Space Administration, Goddard Space Flight Center in Greenbelt, Maryland. The purpose of the proposed project is to implement an automatic system for computer program documentation. The system will produce timely, up-to-date documentation at relatively low cost. The system will be designed to document any computer language and run on any hardware while taking advantage of existing documentation aids. The system will be easy to use and will place minimum restrictions on the programmer.

In computer program development, one of the major problems contributing to low programmer productivity is poor communications. The automatic documentation system will produce documentation at all phases of system design and implementation thereby reducing the communication problem.

The proposed project will span twelve months. The twelfth month will be reserved for installing the automated documentation system at Goddard Space Flight Center in Greenbelt, Maryland. The program is divided into five tasks. The goal of Task 1 will be to describe in detail all of the programs that will be written. Since many of the program requirements are already known, Task 2, writing the programs, can begin concurrently with Task 1 and will take about eight months. System integration will begin at the end of the third month and will be Task 3. Task 3 will include the program checkout phase and should cover about seven months. Once the system is in operation at Texas A&M University, Task 4, installation of the system at Goddard Space Flight Center in Greenbelt, Maryland, will begin. Task 5 commences at the end of the second month

when the detailed design is being phased out. This task will be the continuation of the evaluation and planning phase of the automatic documentation project. At the end of twelve months, personnel working on Task 5 will have evaluated the existing documentation system and proposed a follow-up system for extending and adding capability to the automatic documentation system. Documentation of the automatic documentation system will be delivered along with the final report at the end of the twelfth month.

The Data Processing Center along with the Computer and Information Sciences Division of the Industrial Engineering Department possesses the resources, know-how, and interest to successfully conduct this program. Dr. D. B. Simmons, Director of the Data Processing Center at Texas A&M University, will act as principal investigator. He is uniquely qualified to oversee the implementation of an automatic system for computer program documentation. The Data Processing Center has experience with implementing and operating computer software systems of all sizes and complexity. All necessary hardware and manpower resources will be made available to successfully complete the proposed project. Dr. Simmons has experience in all levels of computer hardware and software design. While an officer in the Signal Corps he evaluated both hardware and software for the U.S. Army. As a member of the staff at Bell Telephone Laboratories he designed and implemented a design automation system for electronic switching systems. One of the main by-products of the system was automated documentation. Also while at Bell Telephone Laboratories, Dr. Simmons designed and implemented the FLARE automatic flowcharting system. This system reduced the cost of producing a flowchart page at Bell Telephone Laboratories from $75 to 50¢. At present this system is the primary flowcharting system used to document programs for electronic switching systems.

The Data Processing Center and the Computer and Information Sciences Division believe that the tasks and systematic approach presented in this proposal are the steps necessary to develop a cost-effective user-oriented automatic documentation system. The proposed system will offer many capabilities that do not exist in any similar system today.

2.0 SYSTEM DESIGN

Under sponsorship of NASA contract NAS5-11911, a detailed survey of all existing documentation aids was conducted. Initially all computer-oriented literature was searched for anything related to documentation or documentation aids. An automated bibliography was used to record all references and key words relating to documentation areas. While documentation is one of the critical areas in software development, literature describing documentation is sparse.

A fruitful source of information about documentation aids was literature supplied by software houses that sell proprietary systems. Every organization that could be found advertising proprietary software aids was sent a letter requesting information. The surprising thing from the information received about the proprietary systems was the lack of features offered by them. Fairly unsophisticated systems are marketed at relatively high prices. No organization has implemented a comprehensive system that covers the whole spectrum of documentation aids. To use existing systems, users are usually required to become expert in numerous special procedures and conventions.

The proposed system will document programs written in FORTRAN, COBOL, PL/1, and Assembly language for the IBM 360 series. It can be easily expanded for use with the Univac 1108 and the CDC 6600 series computers. The various

operating systems were examined and evaluated for similarity. Working papers summarizing the conclusions found in the area of flowcharts, decision tables, operating systems, text editors, program editors, documentation standards, and computer listing formats were produced.

Audio techniques are valuable for capturing documentation information. An experiment was conducted to determine if audio techniques would be useful as an alternative to written text. While it was found that audio techniques have special areas of usefulness, no conclusive evidence was found to justify replacing other techniques with audio documentation.

Following the evaluation of existing documentation aids, a comprehensive automatic documentation system was designed. Details of the design will be presented in the following section. A preliminary design was presented to the NASA representatives at Ames, California during August, 1972. Suggestions made by NASA were incorporated into the design. As a result of the August meeting, a typical program was chosen to demonstrate the type of documentation that would be produced from the automatic documentation system.

3.0 AUTOMATIC DOCUMENTATION SYSTEM DESCRIPTION

3.1 SYSTEM FEATURES

The automatic documentation system will have the following features:

1. <u>Minimal programmer restrictions</u> - For the programmers who write programs in their own unique way, the automated system will be able to produce documentation such as detailed, detail-suppressed, and global flowcharts, data layouts, overlay descriptions, etc. Special cross-reference glossaries can also be produced. Those who use the automatic documentation system during initial project

phases will automatically obtain extensive documentation. But the system will also be designed so that if a program is developed outside the system it will be fairly easy to retrofit the program into the system for documentation maintenance.

2. Eliminate all redundant effort - Documentation produced in an early phase of the design or implementation process can be re-used later on in the development process. For example, if the designer of a programming system describes in detail the function of each program subroutine that is produced, the programmer need not redescribe the function of the routine when he writes his program. The comments describing the purpose of the subroutine will be automatically inserted in the programmer's subroutine.

3. No operating system modification - No justification can be found for modifying an operating system to obtain documentation. Most items of documentation interest can be obtained by scanning output produced by operating systems. Information available in internal tables of an operating system can be reproduced with less effort than it would take to make and maintain modifications to the operating system.

4. Use existing documentation aids - Many man-years of programming effort have gone into developing existing documentation aids. A number of flowcharters already exist. Text editors containing sophisticated algorithms for hyphenation and text layout have been developed. Existing documentation aids will be used as modules in the comprehensive automated system.

5. Interactive/batch system - Users will be able to make use of the system by using batch or interactive systems. The data base

management system that will be used can accept data using either an interactive or a batch mode. The interactive mode will have special features for interrogating the user of the system to obtain necessary documentation data. In the batch mode, the user will supply necessary information using key word or positional parameters. The most user-oriented technique would be the interactive version.

6. <u>Documentation during development</u> - The documentation data base will be constructed from information gathered during all design phases. For example, background or design philosophy sections generated for system specifications can also be used in final documentation. System flowcharts and block diagrams entered during system design can be retrieved from the data base to produce final documentation. Such things as title of program, person responsible for the program, function of subroutine, etc. prepared during the design phase would not have to be re-created during the program implementation phase. The users would supply only that information not available from a previous stage.
7. <u>Accept any language</u> - The automatic documentation system will be language independent. Initially the system will be designed to accept FORTRAN, COBOL, PL/1, and Assembly languages. There will be no restrictions inherent to the documentation system design that will prevent it from being used to document other languages.
8. <u>Operate on any hardware</u> - Initially the system will be designed to operate on the IBM 360 with planned expansion to the Univac

1108 and the CDC 6600 computer systems. These are the three major systems used by NASA. All programs written for the automatic documentation system will be written in a machine independent language so that the automatic documentation system can be easily moved from computer to computer.

9. Monitor and control project - Features will be designed into the system to allow the project manager to monitor the exact status of program development and documentation. In addition, system access and security will be under his control.

3.2 INITIAL PHASE

Work covered by this proposal is termed the initial phase of development of the automatic documentation system. The system will be designed to allow new features to be added at a later date. During the initial phase either FORTRAN or COBOL will be used to write programs that make up the automatic documentation system. Initial development will be for a system operational on an IBM 360 computer, but software will be easily transportable to the Univac or CDC systems. The automatic documentation system will document COBOL, FORTRAN PL/1, or Assembler language programs.

The development process can be segmented into many phases. One way of subdividing the process is shown in Figure 1 where monitor, specifiers, designers, programmers, and validators participate in the development of computer software. From the very beginning of the program development process, the monitor can determine exactly what can be inserted into the data base. The specifier writes the Request for Proposal (RFP) or outlines a need for a program to be developed. He can enter such items as title, abstract, system specifications, testing criteria, block diagrams, and other elements of the RFP. All of

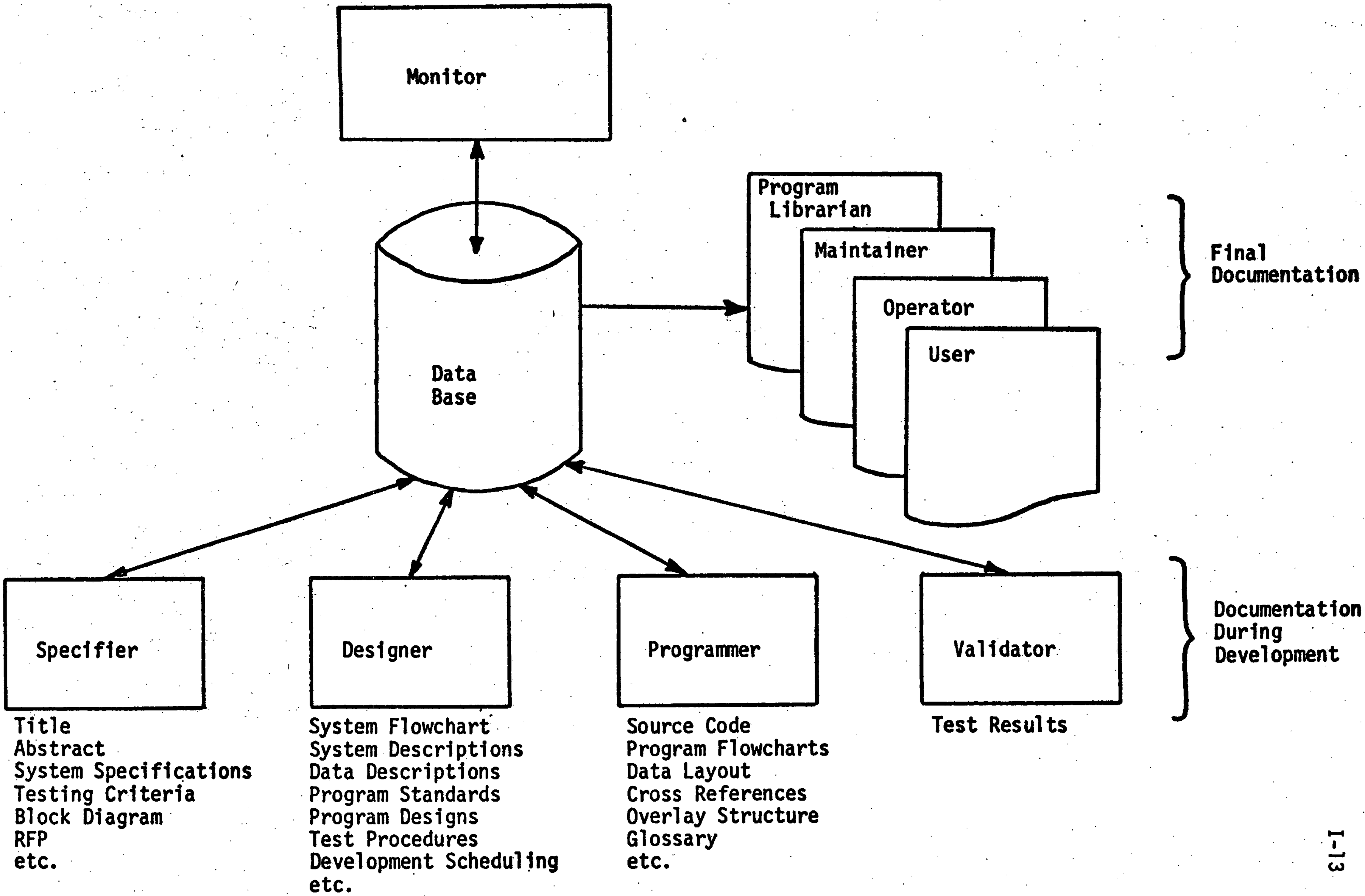


Figure 1 Program Development Process

this information will be stored in the documentation data base. In addition to documentation information, the data base will contain the program source, object and job control modules. In other words, the data base will contain all information necessary to describe and develop the program.

From global specifications, such as the RFP, the designer would draw a detailed system flowchart, produce system descriptions, data descriptions, program standards, subprogram design, test procedures, and produce a development schedule. All design information will be stored into the data base. Documentation aids such as text editors, will minimize the designer's effort related to writing and rewriting his design documents. Information such as background material and subroutine descriptions will be carried through to final documentation. Important design philosophies would be captured at early stages and retained.

Once the design is complete, the programmer can write source code for the program and store it in the data base. From source code and information in the data base such things as program flowcharts, data layouts, glossaries, overlay structures, and extensive cross-reference information can be produced. The monitor program will assist the project manager in determining whether a programmer is conforming to documentation standards and using good programming practices. The monitor can determine the status of the project. Access to different modules will be controlled by the program monitor. Documentation produced during the development process will aid programmers in understanding the operation of other parts of a programming system. This is a major step forward in improving communications between programmers and greatly improves programmer productivity.

Once a set of programs has been completed, a validator can verify that they work. All information necessary for the validation process will have been established by either the specifier, designer, or programmer.

Once the development and implementation of a program has been completed, documentation can be produced for the operation and maintenance phases. The automatic documentation system will produce such things as a user, operator, and maintenance manuals and abstract information for user libraries.

3.3 SYSTEM STRUCTURE

The structure of the proposed automatic documentation system is shown in Figure 2. Three types of programs will be used in the system. Types 1 and 2 are new programs to be developed. Type 3 are existing programs that can be used without change. Type 3 programs make up a major part of the software necessary to implement the automatic documentation system. New programs will not be developed where operational documentation aids are available. By doing this, a sophisticated system will be developed at a relatively low cost.

Type 1 programs make up the executive program routines which will constitute a small part of the overall system. The executive will control access to the system, do syntax analysis on the basic commands (both batch and interactive), gather usage statistics which are not a function of the data base or a single application program and initiate Type 2 programs. Type 1 functions are shown in Figure 3. The major development effort will be concentrated on Type 2 programs.

Type 2 programs that will be developed are listed in Figure 4. A generalized editor will take the place of the normal program editor and do preprocessing

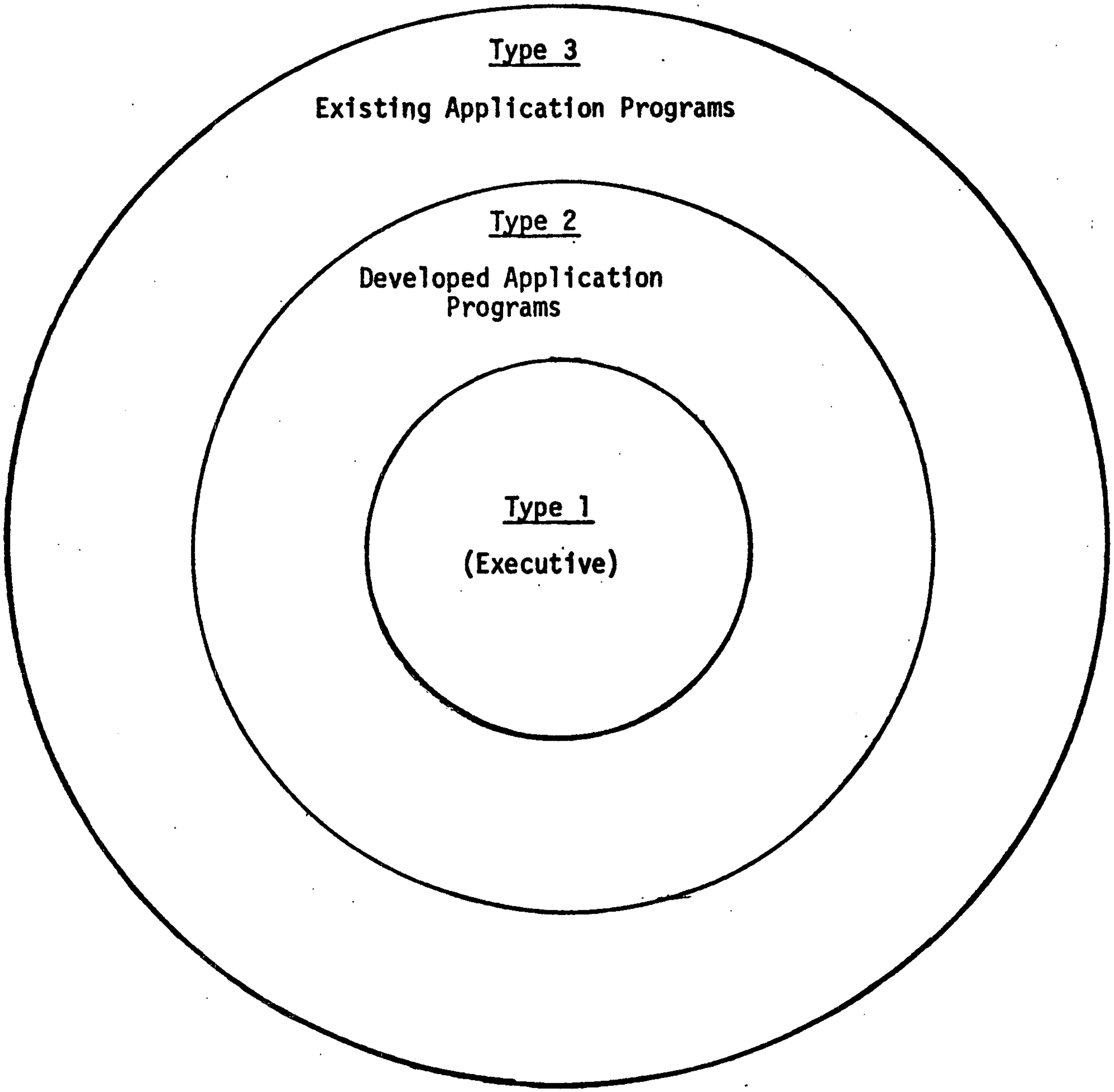


Figure 2 Automatic Documentation System Structure

EXECUTIVE

1. System Access Security
2. Command Processor (Batch/Interactive)
3. Use Statistics
4. Type 2 Initiators

Figure 3 Type 1 Programs

DEVELOPED APPLICATION PROGRAMS

1. Monitor (Control & Statistics)
2. Data Collection
3. Interrogators
4. Batch Command Processors
5. Template Builders
6. Recipe Builders
7. Recipe Scanners
8. Drivers

Figure 4 Type 2 Programs

for a text editor. The reason for using a preprocessor is to facilitate the use of many different text editors. The text entered by the user will be put into a canonical form which can be transformed for use with any proprietary text editor that the user may choose. The editor will also have the ability to insert text information into a program listing without altering the program or data structure.

A monitor program will be developed which will contain control features and produce necessary statistics. The monitor program will check the program listing to insure that proper standards and programming practices have been used. The documentation data base will be checked and statistics produced identifying the types and amounts of data supplied and items that are missing. Exact project status can thus be determined. Documentation error analysis information and access to all data base and program modules will be controlled by the monitor.

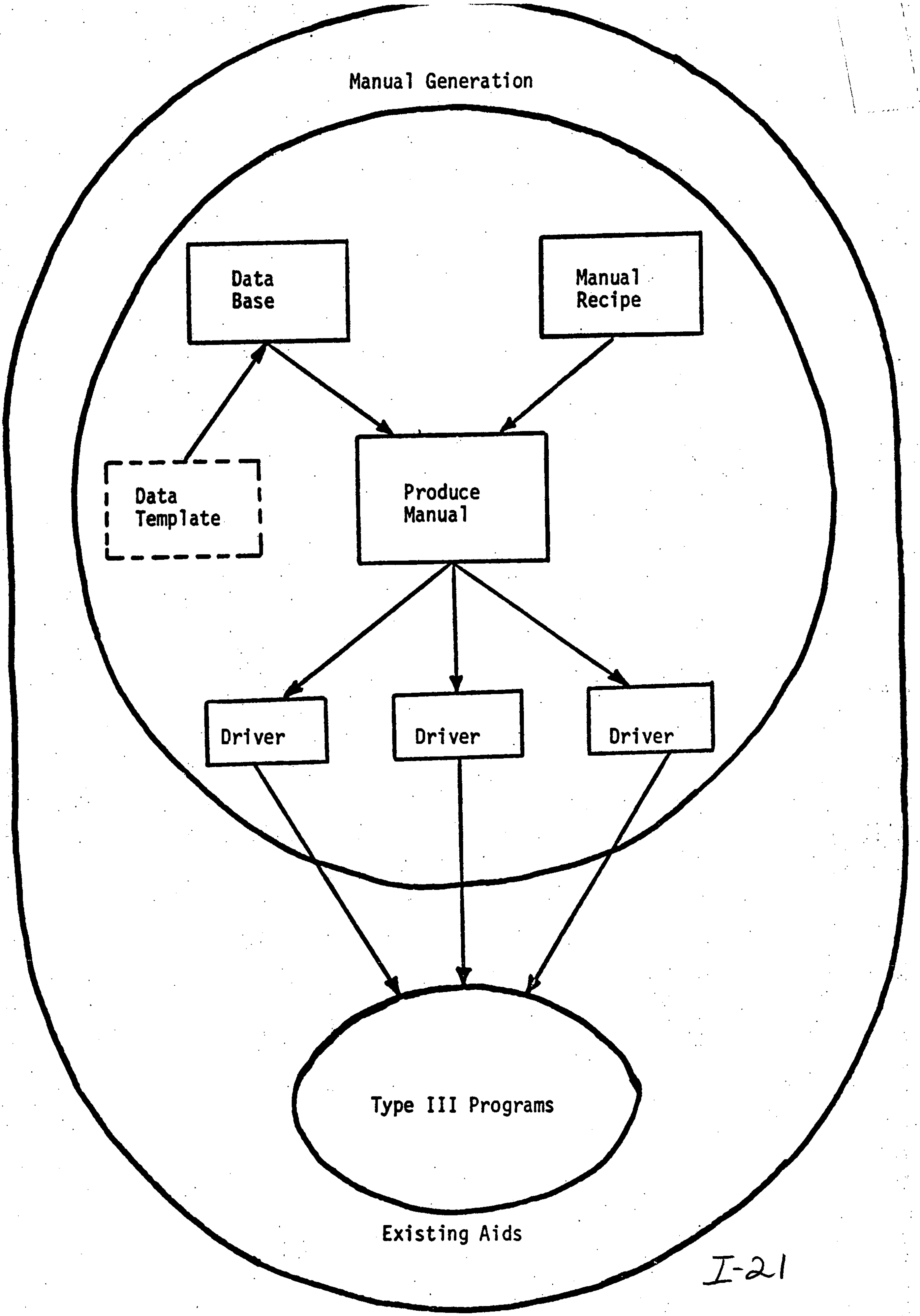
Data collection routines construct the data base. Information supplied by the programmer will be inserted into the data base by data collection programs in a manner transparent to both language and hardware. A standard data base management system will be used, greatly reducing the amount of programming effort necessary to implement the automatic documentation system.

The interactive version of the automatic documentation system will have interrogators for prompting users. In the batch system, a command processor will be necessary to perform the same function as the interactive interrogator system.

During the specification, design, programming, and validation process appropriate information will be stored in the data base. Documentation required during development can then be produced automatically. Data templates

will define information to be stored in the data base. To produce a manual, a recipe must be defined. This recipe consists of items describing exactly what type of information is required in the manual. Recipe building programs will specify the structure for final documentation reports such as user, operator, or maintenance manuals. Recipe scanner programs will then scan these recipes and drive the necessary programs to produce a complete manual. Templates for the data base and recipes for final documentation will be under the complete control of the manager of a programming project. He can decide exactly how much documentation and what type of documentation is to be produced. When the manager does not want to worry himself with format details of final documentation, standard templates of the data base and recipes for the manuals will be supplied by the system. Recipe processing programs will scan the recipe and drive the programs necessary to produce a complete manual. Drivers are called by the recipe scanner that drive application programs.

Relationships among program modules are shown in Figure 5. For example to produce a manual, a recipe for the manual must be defined. Text information entered during the specification or design phases may be needed in the user's manual to describe philosophy behind program development. These items could be specified as part of the recipe. At the point in the manual that the system flowchart appears, the flowchart will be produced automatically from the job control language. To produce the manual, the recipe will be scanned and the appropriate drivers called. To produce the text part of a manual, a driver will take text from the data base and drive a text editor. Using this technique, a different text editor can be used for each different computer. For example, the text editor used for the IBM system might be the TEXT360 or ATS system while for the CDC system, the text editor EDIT, under the KRONOS operating system may be used. Drivers will be tailored to drive available



I-21

Figure 5 Production of Documentation

flowchart systems. to produce flowcharts of programs in various languages.

The system will be able to produce documentation which is not currently available from any existing system. Figure 6 shows levels of documentation that can be produced by the system. Most proprietary systems produce documentation at the local level. They typically produce listings, flowcharts, and some type of cross reference. Some, but not all, systems produce detail suppressed flowcharts. A few are able to produce functional flowcharts. Data layout descriptions usually must be prepared by hand. The proposed system will be able to give a two-dimensional detailed data description and a detail suppressed description. In the future functional data layouts will be automatically produced.

The local level of documentation is associated with a single compilation or assembly. For a program composed of a number of separate modules, global flowcharts,and overlay diagrams are useful. The proposed system will be able to produce these automatically. Global glossaries, cross-reference tables, and data layouts will also be produced. Text will be produced at all levels of documentation. Global flowcharts will be produced from load modules and system flowcharts will be produced from the job control language. System block diagrams can be stored in graphic format and reproduced by the documentation system. All levels of documentation are necessary to understand a complex program. Existing commercial systems give only a small fraction of the necessary information.

The Type 3 programs which will be utilized by the system are shown in Figure 7. These programs will be utilized without modification. For the IBM 360 version, flowcharting systems such as OS/360 Flowchart, QUICKDRAW, or AUTOFLOW will be used. A number of text editors are also available. For the initial phase, the TEXT360 text editor will be used. Output from operating

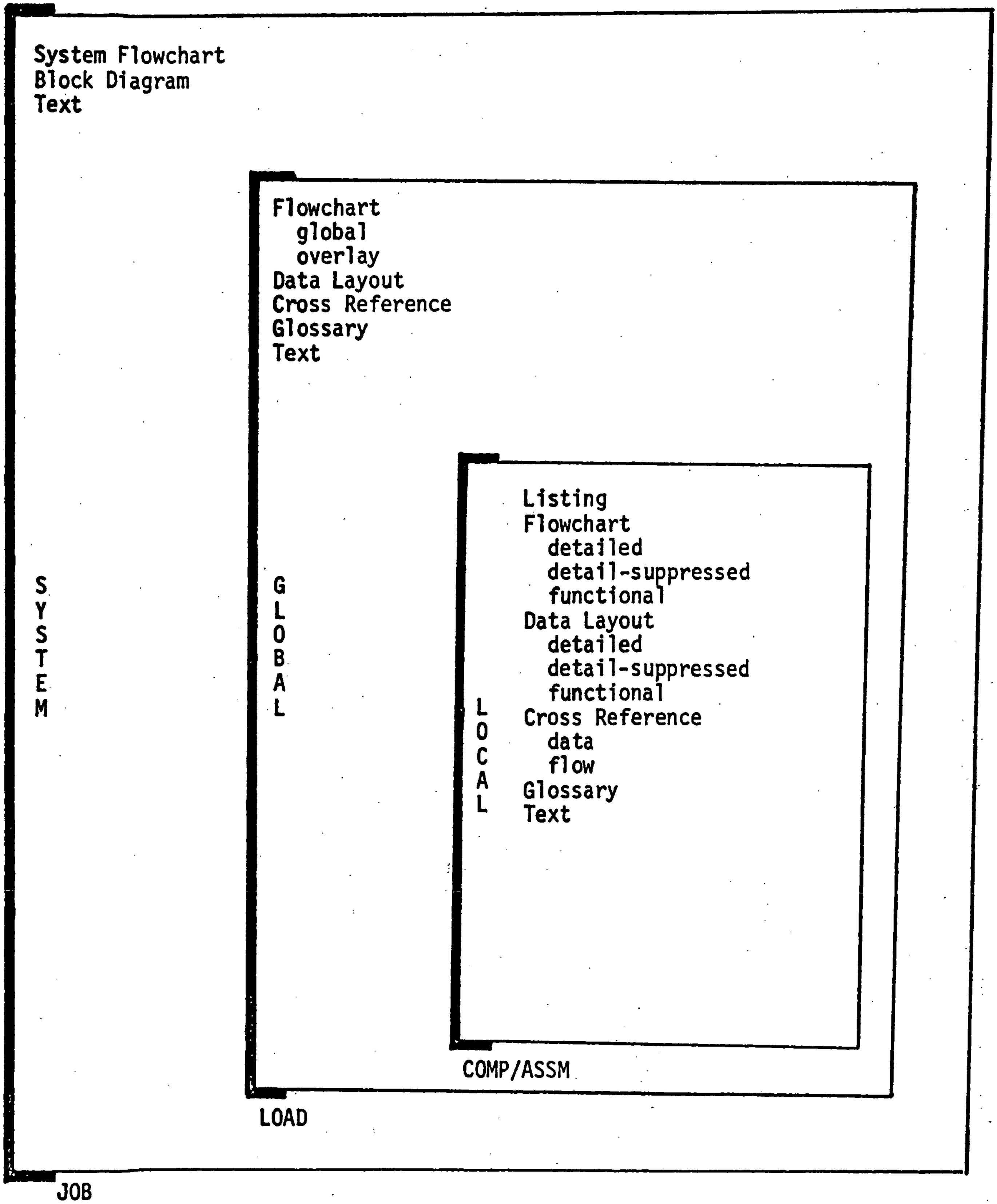


Figure 6 Levels of Documentation

EXISTING APPLICATION PROGRAMS

1. Flowcharters
2. Text Editors
3. Program Editors
4. Index Generators
5. Compilers
6. Tidy Programs
7. Utilities
8. Linkage Editors/Loader
9. Data Base Management
10. Descriptive Statistics

Figure 7 Type 3 Programs

systems and compilers will be used as input to programs that draw cross-reference tables and flowcharts. Therefore, compilers, linkage editors, and loaders are Type 3 programs. Another useful Type 3 program is the TIDY program which does such things as renumber FORTRAN statements in ascending order.

The key to development of the automated documentation system will be use of a generalized data base management system. The System 2000 data base management system developed by MRI Systems Corporation of Austin, Texas will be used. System 2000 is a general purpose data management system with features that include a report writer, a user-oriented language providing on-line access to non-programmers, a procedural language interface for programmer use, sequential file processing, two teleprocessing monitors, and a multiple thread feature. The system provides archival copies of data bases and records an audit trail of changes made to the data base. It is capable of reconstructing a data base by applying an audit trail, completely or in part, to an archival copy of the data base. The procedural language features enable users to manipulate data from COBOL, PL/1, FORTRAN, or Assembler language programs. The data base can be accessed from model 33/35 teletypewriters, IBM 2741 hard copy terminals, and IBM 2260 CRT terminals.

System 2000 provides a wide range of features to insure data base integrity. Five levels of security are provided. Security can be provided at remote terminals through the use of terminal identification or passwords. Password security is available at the system, data base, command, and component level. Component level security enables the data base administrator to offer four types of access for each component in the data base definition. Thus, for each component in the data base there exist sixteen possible access combinations.

4.0 ADVANTAGES

Use of the proposed automatic documentation system offers many advantages over other techniques for developing programs and producing documentation. The system will be user-oriented and will be as easy to operate as existing on-line or batch program editors. Programmers who use good programming practices when developing software can use all of the system features without extra effort. Programmer productivity will be enhanced by improved communication during the development process. The modular system will allow new types of documentation to be easily added to the system. This is the first system that brings together all types of documentation aids into a single user-oriented system. It emphasizes documentation on a load module and system basis as well as for a single compilation. Documents can be produced that are made up of heterogenous output such as text and flowcharts. Managers can use the system to control and monitor projects. Program and documentation standards can be enforced and taught through the use of the automatic documentation system. People who do not use the documentation system during program development will be able to use it for post-development documentation. Programs developed independently of the automatic documentation system can easily be retrofitted into the documentation data base.

5.0 FUTURE EXPANSION

As previously mentioned, the system has been designed in a machine independent and language independent manner so that it can be easily moved from machine to machine and used to document various languages. During the initial phase the only graphics will be flowcharts. Graphic capabilities can be added to the system at a future time. Any type of two-dimensional drawing

Months	0	1	2	3	4	5	6	7	8	9	10	11	12
Detailed Design													
TASK 1	Detailed Design												
TASK 2	Program Implementation												
TASK 3					Program Checkout & Test								
TASK 4										Install System at Goddard			
TASK 5				Study and Evaluation									
Detailed Design					X								
Progress Reports		X	X	X	X	X	X	X	X	X	X	X	X
Briefings	()				()				()			()	
Final Report													()

Figure 8 Program Tasks

could then be placed in a data base. Provision for drawing sophisticated data layout descriptions can be added to the system. Documentation of data bases and complex data structures can also be automatically drawn. During the initial phase the system will be designed to run on the IBM 360, but it can easily be expanded to systems such as the Univac 1108 and CDC 6600 that have FORTRAN or COBOL compilers. A syntax-directed language extension program can be developed to facilitate addition of new languages and output devices.

Long-term future developments could be to look into the possibility of producing all types of documentation from programs that contain unstructured comments. A number of program ambiguities can be resolved by simulating program execution. Management tools such as PERT diagrams of project status can be automatically produced from information found in the documentation data base.

6.0 PLAN OF ACTIVITY

The implementation of the automatic system for computer program documentation will be divided into five tasks as shown in Figure 8. The first task will be to design each program in detail. The second task will be to write the programs. The third task will be to test the programs. The fourth task will be to install the working system at the Goddard Space Flight Center. The final task will be to continue the study and evaluation of documentation aids to extend the initial phase of the automatic documentation system development.

6.1 TASK 1 - DETAIL DESIGN

While the design of the overall system will be completed before the projects starts, the detailed design of each routine of the system will take place during the initial development process. A design document will be

produced as a result of Task 1. Final documentation describing the automatic documentation system will be produced by the automatic documentation system.

6.2 TASK 2 - PROGRAM IMPLEMENTATION

Task 2 will be the major program implementation phase. Programming will fall into four major areas: (1) monitor program, (2) template, recipe, and data base builders and recipe scanner, (3) program editor, text editor driver, routines for storing FORTRAN, Assembler, COBOL, and PL/1 source programs and routines for using a language processor and operating system as input (4) drivers to produce local, global, and system documentation. A programmer will be assigned to each of these major areas. In addition, a number of modules for producing different types of documentation will be produced by graduate students as part of their normal course work at no cost to NASA.

6.3 TASK 3 - PROGRAM CHECKOUT AND TEST

The automatic documentation system will document the following programs: (1) All Type 1 and Type 2 programs of the automatic documentation system, (2) A scientific program written in FORTRAN, (3) Administrative programs developed by the Texas A&M University Data Processing Center. Checking and debugging of the system will be done by the programmers assigned to software development. Exercising of the system will be done by graduate assistants assigned to the project. Members of the Data Processing Center staff who use the automatic documentation system to document their programs will not charge their time to the project. Computer costs resulting from documentation produced for the Texas A&M University Data Processing Center will not be charged to the project.

6.4 TASK 4 - SYSTEM INSTALLATION AT GODDARD

The system will be installed at Goddard Space Flight Center after it has been extensively checked out at Texas A&M University. Therefore, installation and on-site evaluation should take less than a month. Project evaluation will be conducted during this period.

6.5 TASK 5 - STUDY AND EVALUATION

Study and evaluation of all types of documentation aids will continue. The bibliography on automatic documentation will be updated and maintained. Personnel assigned to this task will do an on-going evaluation of the automatic documentation system. Future extensions will be designed by this group.

7.0 DATA DOCUMENTATION AND REPORTS

Texas A&M University will furnish the National Aeronautics and Space Administration the following items.

7.1 PROGRESS REPORT

A monthly progress report covering progress made during the previous month shall be delivered to the Contracting Officer by the tenth of the month following the month reported. This report shall state, in concise terms, items such as accomplishments, estimates of funds, commitments during the reporting period, plans for the next period, problems and anticipated delays, and specific recommendations to facilitate execution of the contract.

7.2 BRIEFING

During the first month after contract award, a meeting with the Contract Officer and other interested NASA officials in Maryland can be held to discuss

the detailed design. At this meeting, Texas A&M will discuss planning for project activities. At the end of the fourth month, the detailed design will be completed and can be presented to the Technical Officer and other members at NASA. If deemed appropriate, another briefing can be given at the end of the eighth month. At the end of the project, a briefing can be given to describe what has been accomplished and suggest system extensions.

7.3 DETAILED DESIGN

At the end of the fourth month the detailed design will be completed and a written report will be produced. This design document will be fed into the data base as an integral part of the documentation to be produced for the project.

7.4 DELIVERABLE ITEMS

Source and object programs will be delivered to NASA in Greenbelt, Maryland and installed on an IBM 360 computer. Complete documentation of the programs will be supplied. All of the documentation will be produced using the automatic documentation system. Final documentation produced will be an operator's manual, user's manual, and maintenance manual.

7.5 FINAL PROJECT REPORT

The contractor shall furnish a draft of the final project report within 338 days after the effective date of the contract. The government shall be allowed seven days to review and return the report with comments and recommendations to the contractor. The comments and recommendations of the Contract Manager shall be taken into consideration in preparing the final report which will be submitted to NASA within 21 days after the Contract Officer approval of the date. The final project report will completely document all research,

recommendations, and results of the efforts during the performance of the contract.

8.0 PROGRAM ORGANIZATION

The proposed program will be conducted by members of the professional staff of the Data Processing Center and of the Computer and Information Sciences Division within the College of Engineering at Texas A&M University. Dr. D. B. Simmons, who is Director of the Data Processing Center and a member of the Computer and Information Sciences Division faculty will act as principal investigator. The Data Processing Center facilities are designed to accomodate the teaching, research, and administrative needs of the university. Approximately 50% of the machine time is used by research projects, 20% by academic efforts, and 30% for administrative processing. The Data Processing Center employs approximately 80 people, including 33 professionals. Personnel are organized into seven groups: System Software, Computer Operations, Computer Systems, Administrative Applications, Agricultural/Statistics, Office Operations, and Fiscal. The Data Processing Center operates as a separate entity with a 72-73 Fiscal Year budget in excess of $1.5 million.

The Computer and Information Sciences Division is part of the Industrial Engineering Department within the College of Engineering at Texas A&M University. The Industrial Engineering Department has one of the best programs in the country maintaining an effective balance between teaching and research. The Computer and Information Sciences Division was established in 1963 and its growth is best indicated by its size:

(1) 16 professional staff members - 9 with Ph.D.'s and 7 with Ph.D.'s in progress.

(2) 148 graduate students, 112 at the Master level and 36 working toward a Ph.D.

(3) a new undergraduate program in Computing Science will start in the late fall of 1972.

9.0 PERSONNEL QUALIFICATIONS

The Data Processing Center along with the Computer and Information Sciences Division possess the resources, know-how, and interest to successfully conduct this project. The following people will work directly on the project: Dr. D. B. Simmons - one quarter time, Dr. R. W. Elliott - one quarter time, Dr. D. Colunga - one quarter time, Ms. S. Arseven - full time, Mr. G. H. Kemper - full time, Mr. M. H. Lyle - full time, and Mr. P. Crews - full time.

Dr. D. B. Simmons, who will serve as principal investigator, has been associated with computer field for over 13 years. He started in circuit design in 1959 on the early RCA semiconductor computers. He served in the Army as Systems Evaluator for computer systems used by the Army. In 1963 he joined Bell Telephone Laboratories where he worked in the areas of logic design, design automation, automatic program documentation, and automatic flowcharting. He designed and developed the FLARE automatic flowcharting system. He served as supervisor of the Advanced Programming and Processor Technology Group which worked on designs for the 1980's, programs for improving design automation systems, high-level electronic switching systems (ESS) languages, and automatic flowcharting systems. Since joining Texas A&M University in 1970, Dr. Simmons has been involved in teaching, consulting, and development of computer operating systems. He has served as principal investigator on a project for the development of an operating system for mini-computers and as principal investigator on the project to design the automatic documentation system for NASA. He is currently Director of the Data Processing Center and Associate Professor of Computing Science.

Dr. Elliott's main area of interest is Computing Science with specialization in computer graphics and information retrieval. Dr. Elliott was Associate Director of the Texas Regional Academic Computing Experiment Project sponsored by the National Science Foundation. This project had regional computing facilities to Tarleton, Prairie View, and Texas Southern Universities from Texas A&M University's computer center. Dr. Elliott has experience in computer programming, consulting in various computer areas, and in teaching computer oriented subjects.

Dr. Colunga has worked with computers since 1958. From that time, he has acquired experience on the following computers: IBM: 650, 1620, 7090, 7094, 360/65; CDC: 1604, 3600, 6600; Univac 1108. Until last year Dr. Colunga was associated with the Theory and Analysis office of the Computation and Analysis Division, under the direction of Eugene Brock at NASA/MSC at Houston, Texas. While there, Dr. Colunga initiated efforts toward the problem of documentation of control-optimization programs available to NASA/MSC users through the program share library facilities coordinated by Mr. John Leonard. Dr. Colunga is therefore aware of the problems associated with NASA's needs for scientific computer program documentation.

Ms. Susan Arseven has been active in the field of Computer and Information Science for more than 8 years. She began her participation in 1964 designing and marketing automated systems for libraries and information centers in New York City for IBM Corporation. In 1967 she went to the University of Pennsylvania as the Head of the System Planning Office at the University Library where she designed and developed an automated circulation control system, a serials catalog, and a book acquisition system and served as advisor to the

Director of Libraries on computers and automated techniques. She has made major contributions to the project to design the automated documentation system for NASA. She is currently a Systems Analyst at the Data Processing Center and an Assistant Professor of Computing Science.

Mr. Kemper and Mr. Lyle have both had extensive experience in systems programming and applications programming at Texas A&M University and in industry. They are currently both senior members of the Systems Programming Group of the Data Processing Center. Mr. Crews, a new addition to the programming staff at the Data Processing Center, has a background in scientific programming and has developed a mini-computer independent macro processor.

Besides the above mentioned personnel, this project will use other members of the Data Processing Center and Computer and Information Sciences Division faculty on a consulting basis. In addition, there are a number of Air Force officers who are working on Master's and Ph.D. degrees in Computing Science. They have extensive experience in application areas of computers within the government and have a first-hand knowledge of documentation problems. Any Air Force officer who works on this project in conjunction with a Master's or Ph.D. dissertation will charge no time to the contract since he is already being fully funded by the government.

In addition to the professional staff, three research assistants will work in a supporting capacity. Also a half-time secretary will support the project. The Data Processing Center and the Computer and Information Sciences Division have the expertise in designing, developing, and coordinating software systems to successfully accomplish the goals stated in this proposal.

10.0 FACILITIES

Texas A&M Data Processing Center is a centralized facility serving the computing needs of the entire Texas A&M University campus. It is responsible for all of the administrative data processing, educational support, and research support on campus.

The principal computer is an IBM 360/65 with 512 thousand bytes of main core storage and 2 million bytes of extended core storage (large core storage). This computer is supplemented by an IBM 7090 system used to control experiments located in the Cyclotron building. In addition, there is a tape-oriented 1401 computer that supports the 7090 computer. There is also a PDP mini-computer supporting the IBM 360/65 computer.

The initial system 360 was installed in December, 1967. The present configuration includes three card reader punches, four printers, seven magnetic tape drives, 24 2314 disk drives, two IBM 1050 communication terminals, four 2740 terminals, five 2741 terminals, and 19 720 Sander Associates terminals. The computer also has remote job entry stations at other locations on campus such as the library, the Engineering Building, and it has entry stations off campus in Houston, San Antonio, Stephenville, and Austin. A CalComp Plotter and a Gerber automatic drafting machine are available for producing two-dimensional graphs.

Besides the main computer center, there are numerous small computers around campus. The Computer and Information Sciences Division has a Data General Nova computer with a magnetic tape unit, high speed paper tape reader, and an interactive CRT. Also the division has an intelligent terminal connected to the main computer.

11.0 PROGRAM SCHEDULE

The program will be divided into five tasks that will cover a period of 365 days. The tasks will be broken down as follows:

TASK ONE - DETAILED DESIGN

Starting Time - Beginning First Month

Duration - 4 Months

Personnel Involved:	Dr. D. B. Simmons	-	1 man month
	Dr. R. W. Elliott	-	3/4 man month
	Dr. D. Colunga	-	3/4 man month
	Ms. S. Arseven	-	1 man month
	Graduate Research Assistant	-	4 man months
	Secretary	-	2 man months

TASK TWO - PROGRAM IMPLEMENTATION

Starting Time - Beginning First Month

Duration - 8 Months

Personnel Involved:	Ms. S. Arseven	-	6 man months
	Mr. G. H. Kemper	-	8 man months
	Mr. M. H. Lyle	-	8 man months
	Mr. P. Crews	-	8 man months
	Graduate Assistant	-	8 man months

TASK THREE - PROGRAM CHECKOUT AND TEST

Starting Time - Beginning of Fourth Month

Duration - 8 Months